# DOCUMENT RESUME

ED 281 023 CE 046 876

| | |
|---|---|
| AUTHOR | Roth, Gene; Tesolowski, Dennis |
| TITLE | Planning, Executing, and Evaluating Competency-Based Instruction. Microcomputer Applications for Vocational Teachers: A Competency-Based Approach--Book C. |
| INSTITUTION | Idaho State Univ., Pocatello.; Illinois State Board of Education, Springfield. Dept. of Adult, Vocational and Technical Education. |
| PUB DATE | Jun 86 |
| NOTE | 104p.; For related handbooks, see CE 046 874-878. |
| PUB TYPE | Guides - Classroom Use - Guides (For Teachers) (052) |
| EDRS PRICE | MF01/PC05 Plus Postage. |
| DESCRIPTORS | Behavioral Objectives; Competency Based Teacher Education; *Computer Assisted Instruction; Computer Literacy; Computer Oriented Programs; Computer Uses in Education; *Courseware; Individualized Instruction; Inservice Teacher Education; Instructional Effectiveness; *Microcomputers; Needs Assessment; Resource Materials; Secondary Education; Student Evaluation; *Teacher Improvement; *Teaching Methods; Units of Study; Vocational Education; *Vocational Education Teachers |

ABSTRACT

This handbook is the third in a series of five competency-based resource guides on microcomputer applications for vocational teachers. The 13 units of instruction in this handbook are concerned with the content of the 14 competencies included in the category, "Planning, Executing, and Evaluating Competency-Based Instruction." Units are designed to prepare teachers to do the following: (1) differentiate among applications of computer-based instruction (CBI), (2) assess students' needs for specific CBI applications, (3) develop lesson plans for incorporating CBI, (4) select appropriate software for specific instructional purposes, (5) modify software, (6) design software, (7) prepare instructional materials to accompany software, (8) modify software documentation for specific instructional use, (9) orient students to CBI, (10) execute CBI, (11) individualize instruction with CBI, (12) assess students' microcomputer skills, and (13) evaluate and modify CBI based on student achievement. Components of each unit include unit and specific objectives, informative material, sample forms and evaluation measures, examples, a summary, achievement indicators, and a list of references. (YLB)

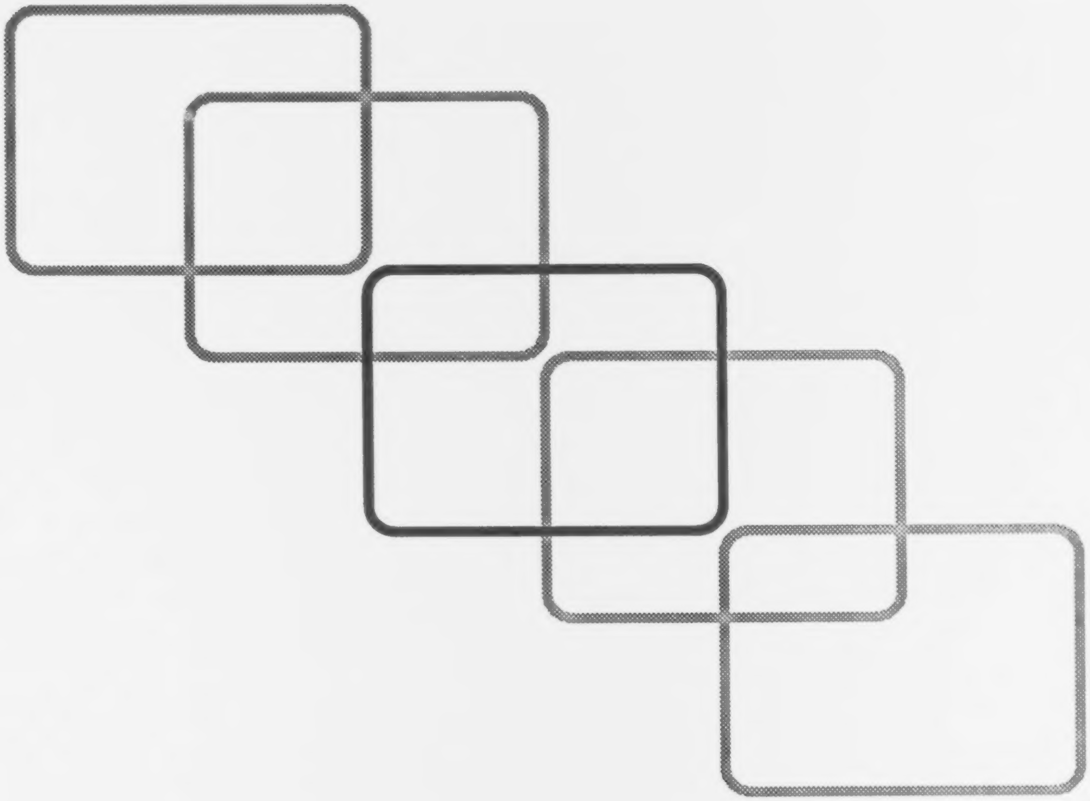
***********************************************************************
*    Reproductions supplied by EDRS are the best that can be made     *
*                   from the original document.                       *
***********************************************************************

ED281023

# Planning, Executing, and Evaluating Competency-Based Instruction

Illinois
State Board of
Education

Adult,
Vocational and
Technical Education



CE046876

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☑ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

- Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

BEST COPY AVAILABLE

# Planning, Executing, and Evaluating Competency-Based Instruction

Project Staff:

Gene Roth, Director

Dennis Tesolowski, Principal Investigator

Department of Vocational Education,
Idaho State University

in cooperation with

Illinois State Board of Education,
Department of Adult, Vocational,
and Technical Education

Illinois
State Board
of Education

Walter W. Naumer, Jr.
Chairman

Ted Sanders
State Superintendent
of Education

Department of
Adult,
Vocational and
Technical Education

Research and
Development Section

June, 1986

# TABLE OF CONTENTS

Acknowledgements .......... C-iii

Introduction .......... C-1

Unit 1
Differentiate Among Applications of Computer-Based Instruction .......... C-5

Unit 2
Assess Students' Needs for Specific Computer-Based Instruction Applications .......... C-10

Unit 3
Developing Lesson Plans for Incorporating Computer-Based Instruction .......... C-13

Unit 4
Select Appropriate Software for Specific Instructional Purposes .......... C-24

Unit 5
Modifying Software .......... C-43

Unit 6
Designing Software .......... C-51

Unit 7
Prepare Instructional Materials to Accompany Software .......... C-57

Unit 8
Modify Software Documentation for Specific Instructional Use .......... C-66

Unit 9
Sink or Swim: Orienting Students to Computer-Based Instruction .......... C-70

Unit 10
Execute Computer-Based Instruction .......... C-75

Unit 11
Individualizing Instruction with Computer-Based Instruction .......... C-81

Unit 12
Assess Students' Microcomputer Skills .......... C-85

Unit 13
Evaluate and Modify Competency-Based Instruction Based on Student Achievement .......... C-88

This publication is available from:

CURRICULUM PUBLICATIONS CLEARINGHOUSE
Western Illinois University
Horrabin Hall 46
Macomb, IL 61455

800-322-3905 (toll free in Illinois)
309-298-1917 (from outside Illinois)

5

# ACKNOWLEDGEMENTS

This handbook evolved from the efforts and creative thought of a very special research team. Harold Blackman, Roger Rankin, and Dennis Tesolowski were the stalwarts of this research endeavor. They comprised the team which nurtured this project from its inception to its completion.

The foundation for this handbook was laid by a panel of Illinois vocational educators. Individuals were selected to serve on this panel on the basis of demonstrated leadership in the use of microcomputers. Utilizing a structured process known as DACUM (Develop A Curriculum), this group developed the initial competency list for the handbook and field tested the product. The DACUM participants included:

James Hamilton, Facilitator
Senior R & D Specialist
The National Center for
Research in Vocational Education
The Ohio State University
Columbus, Ohio

## Panelists

Connie Allekian
River Forest, IL

Betty Bromley
Moline, IL

Catherine Carter
Springfield, IL

Rose Christof
Springfield, IL

Ron Dale
McHenry, IL

Tom Faulkner
Biggsville, IL

Judy Garland
Deerfield, IL

Tom Holloway
Granite City, IL

David Kietzman
Danville, IL

Horace Marvel
Quincy, IL

Robert Meeker
Bloomington, IL

Carleen Presley
Glen Ellyn, IL

In addition to the research team, the following authors wrote units of instruction. Their collective expertise helped shape this document.

Harold S. Blackman
Adjunct Assistant Professor
Idaho State University

Wayne Daw
Instructor, Microcomputer Laboratory
Vocational Technical School
Idaho State University

Bruce DeBoer
Microcomputer Specialist
Moraine Park Technical Institute
Fond du lac Campus
235 North National Avenue
Fond du lac, Wisconsin 54935

Michael T. Drotter
Director, Nu-Ed. Associates
329 G. Mink Creek Road
Pocatello, Idaho 83201

Don Eshelby
Coordinator, Planning and Evaluation
Idaho Division of Vocational Education
Boise, Idaho

Emma Gebo
Assistant Professor
Chair, Department of Home Economics
and Vocational Teacher Education
Idaho State University

Phil Gibson
Manager, Services Division
Vocational Technical School
Idaho State University

Richard A. McEwing
Assistant Professor
Director, Field Experiencecester
Idaho State University

Horace Marvel
Data Processing Instructor
Quincy Area Vocational Center
Quincy, Illinois

Jack A. Piel
Assistant Professor
Director, Early Childhood Education
Idaho State University

Roger A. Rankin
Assistant Professor
Director, Center for Economic Education
Idaho State University

Dennis G. Tesolowski
Associate Professor
Department of Industrial Education
Clemson University
Clemson, South Carolina

Robert Watts
Associate Professor
Director, Campus Computer Centers
Idaho State University

Dr. Molly Wilson-Drotter
School District No. 25
Pocatello, Idaho

## INTRODUCTION

# Microcomputer Applications For Vocational Teachers: A Competency-Based Approach

BY:

DR. GENE L. ROTH

DR. DENNIS G. TESOLOWSKI

Historically, vocational educators have had to cope with the problem of keeping pace with technology. Preparing students for a workplace that is continually changing is a constant reminder to vocational instructors that they do not have the luxury of resting on previously learned work skills and knowledge. Vocational educators must keep abreast of contemporary developments within their vocational area of expertise.

This concern for technical updating is not limited to industrial or business applications of technology. In addition to concerns about preparing students for a changing world of work, vocational teachers must contend with applications of new instructional technologies. Many vocational teachers are currently struggling with how to integrate computer-based instruction into their classrooms and laboratories.

The rapid influx of microcomputers into vocational classrooms and laboratories has caught many vocational educators unprepared to effectively utilize this contemporary instructional technology. As educational systems continue to acquire computer technology, many vocational instructors are saying, or at least thinking, "Where do we start with these machines?" Microcomputers are often purchased for vocational programs which are staffed by personnel that have not been appropriately trained in the technology. Their knowledge of hardware and software may be quite limited. A resulting danger is that microcomputers will be misused or not used at all because vocational teachers have been inadequately acquainted with educational computing (Pratscher, 1983).

This concern about providing vocational educators with pertinent information related to microcomputer applications has brought about a collaborative effort between two state offices of vocational education. The Illinois State Board of Vocational Education, Department of Adult, Vocational, and Technical Education and the Idaho State Board of Education, Division of Vocational Education are jointly supporting this research and development project entitled "Microcomputer Applications for Vocational Teachers: A Competency-Based Approach." This project, which has been conducted at Idaho State University, features a systematic approach to the identification of microcomputer competencies for vocational instructors (Roth & Tesolowski, in press).

---

This is a shortened version of an article that appeared in *The Computing Teacher*, 12 (3), November 1984. Reprinted by permission.

# The DACUM Process: A Method for Identifying Microcomputer Competencies

The DACUM (Developing A Curriculum) process (Adams, 1975) was utilized by this project as a foundation in the development of competency-based materials on microcomputer applications for vocational instructors (Roth, Tesolowski, Rankin, & Blackman, 1984). This procedure is based on three assumptions: (a) expert workers can define and describe their job more accurately than anyone else; (b) any job can be effectively described in terms of the tasks that successful workers in that occupation perform; and (c) all tasks, in order to be performed correctly, demand certain knowledge and attitudes from workers (Miller-Beach, 1980).

Utilization of the DACUM process required the project to assemble a panel of 12 vocational educators. The 12 members, all from Illinois, included 4 secondary vocational instructors, 4 post-secondary vocational instructors, 3 secondary vocational administrators, and 1 representative of the Department of Adult, Vocational, and Technical Education. In addition to being practitioners in the field of vocational education, these individuals have been recognized as leaders in the state of Illinois at applying microcomputers in their work. The challenge for the DACUM panel was to identify competencies specific to the application of microcomputers in vocational education. This was accomplished through a process of competency identification and consensus decision-making. The activity involved the panelists and the facilitator in two days of difficult work. However, the panelists were rewarded for their efforts as competencies were established for each category and the final profile of microcomputer applications for vocation al educators unfolded. Furthermore, the panelists began to realize that they had increased their own personal levels of knowledge about the application of microcomputers in vocational education.

## RESULTS OF THE DACUM PROCEDURE

Most vocational teachers recognize the vast potential of microcomputers in vocational education. However, many professionals have had difficulty identifying the precise role of the machine in their professional lives. The DACUM profile provides teachers with a graphic portrayal of how the microcomputer integrates with the overall schema of vocational instruction and curricula. The profile consists of 47 competencies clustered within the following 5 categories (Table 1):

A. Developing a personal plan for microcomputer competency.

B. Integrating computer-based instruction (CBI) into vocational curricula.

C. Planning, executing, and evaluating CBI.

D. Planning and organizing vocational education learning environments for CBI.

E. Performing classroom management functions with CBI.

# VOCATIONAL TEACHER COMPETENCY PROFILE FOR MICROCOMPUTER APPLICATIONS

**Illinois State Board of Education**
**Department of Adult, Vocational and Technical Education**
**Idaho State Board of Vocational Education**
**Division of Vocational Education**

**Dr. Gene L. Roth**
**Project Director**
**Dr. Dennis G. Tesolowski**
**Dr. Roger A. Rankin**
**Dr. Harold S. Blackman**

◄ **Competencies** ►

| | **Category** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **A** | **Developing a Personal Plan for Microcomputer Competency** | A.1 Define Elements of a Local Education Agency (LEA) Plan for Computer Based Instruction (CBI)<br>3.67 | A.2 Define the Vocational Instructor's Role in the LEA Plan for CBI<br>4.00 | A.3 Conduct a Personal Assessment of Microcomputer Competency<br>3.90 | A.4 Set Personal Goals for Microcomputer Competency<br>4.11 | A.5 Construct a Personal Plan for Microcomputer Competency<br>4.06 | A.6 Implement a Personal Plan for Microcomputer Competency<br>4.16 | A.7 Evaluate Personal Plan Based on Computer Innovations<br>3.87 | A.8 Modify Personal Plan as Needed<br>3.97 |
| **B** | **Integrating CBI into Vocational Curricula** | B.1 Identify Job-Specific Applications of Microcomputers for inclusion in Vocational Curricula<br>4.42 | B.2 Identify Educational Applications of Microcomputers for inclusion in Vocational Curricula<br>4.37 | B.3 Develop a Plan to Apply CBI to Vocational Curricula<br>4.38 | B.4 Demonstrate an Awareness of Microcomputer Software for Developing Computer Based Vocational Curricula<br>4.22 | B.5 Write Specifications for Hardware/Software Based on Vocational Curricula Requirements<br>3.77 | B.6 Implement a Plan to Apply CBI to Vocational Curricula<br>4.26 | B.7 Evaluate Applications of CBI in Vocational Curricula Based on Innovation in Computer Technology & Work<br>3.95 | B.8 Modify Applications of CBI in Vocational Curricula as Needed<br>4.04 |
| **C** | **Planning, Executing & Evaluating CBI** | C.1 Differentiate Among Applications of CBI (Such as Drill & Practice, Tutorial, Simulation & Problem Solving)<br>3.84 | C.2 Assess Students Needs for Specific CBI Applications<br>4.15 | C.3 Develop Lesson Plans Incorporating CBI<br>4.22 | C.4 Select Appropriate Software for Specific Instructional Purposes<br>4.61 | C.5 Modify Simple Software for Specific Instructional Purposes<br>3.71 | C.6 Develop Software for Specific Instructional Purposes<br>3.38 | C.7 Prepare Instructional Materials to Accompany Software<br>4.22 | C.8 Modify Software Documentation for Specific Instructional Purposes<br>3.77 |
| | **Planning, Executing & Evaluating CBI (Con't)** | C.9 Orient Students to CBI<br>4.43 | C.10 Execute CBI<br>4.39 | C.11 Individualize Instruction with CBI<br>4.17 | C.12 Assess Student Microcomputer Skills<br>4.02 | C.13 Evaluate Effectiveness of CBI Based on Student Achievement<br>4.21 | C.14 Modify CBI as Needed<br>4.10 | | |
| **D** | **Planning & Organizing the Vocational Education Learning Environment for CBI** | D.1 Develop a Plan to Implement CBI in Vocational Education Learning Environments<br>4.11 | D.2 Schedule CBI Activities<br>3.97 | D.3 Project Resource Needs (Supplies, Materials, Equipment) for CBI<br>4.02 | D.4 Provide Microcomputer Maintenance<br>3.69 | D.5 Establish Microcomputer User Security<br>3.72 | D.6 Establish Microcomputer Hardware/Software Security<br>3.84 | D.7 Create Authorized Back-up Copies of Microcomputer Software<br>4.40 | |
| **E** | **Performing Classroom Management Functions with CBI** | E.1 Determine Classroom Management Activities to be Performed with CBI<br>4.13 | E.2 Select Software for Classroom Management Activities<br>4.28 | E.3 Prepare Software for Classroom Management Activities<br>3.57 | E.4 Maintain Classroom Rosters<br>3.44 | E.5 Maintain Attendance Records<br>3.39 | E.6 Generate Tests<br>3.83 | E.7 Score Tests<br>3.53 | E.8 Record Grades or Performance Progress<br>3.88 |
| | **Performing Classroom Management Functions with CBI (Con't)** | E.9 Maintain Inventory of Teaching Supplies, Materials & Equipment<br>3.91 | E.10 Manage Vocational Student Organizations<br>3.49 | | | | | | |

**Competency Rating Scale**

The relative importance of these 47 competencies was determined by surveying a national sample of 134 vocational educators. Vocational instructors included in this sample were identified as experts at applying microcomputers in their programs by their respective state supervisors. Ninety-seven (97) vocational teachers (72%) responded to this survey. The following vocational disciplines were represented by this sample: agriculture, business, home economics, marketing and distribution, trade and industrial, and health occupations education. Mean ($\bar{x}$) compentency ratings were derived from respondents' ratings based on the following scale.

**1 = No Importance** **2 = Minimal Importance** **3 = Average Importance** **4 = High Importance** **5 = Extreme Importance**

The content of these 47 competency statements was refined and validated through a formative process. After the DACUM panel had generated the core of this profile, the competency statements were scrutinized and revised by: (a) members of the project team at Idaho State University; (b) a group of vocational educators in Idaho; (c) consultants of the Illinois Department of Adult, Vocational, and Technical Education; and (d) supervisors and staff members of the Idaho Division of Vocational Education.

A survey was conducted by this project's research team to ascertain the relative importance of each of the 47 microcomputer competencies. The survey population consisted of a national sample of 134 vocational educators. These instructors were identified by their respective state supervisors as leaders in their states at applying microcomputers to the roles and responsibilities of their teaching jobs. Ninety-seven vocational teachers (72%) responded to the survey.

Ratings for each competency are listed on the *Vocational Teacher Competency Profile for Microcomputer Applications* (Table 1). Mean (x) competency ratings were derived from respondents' ratings on the following scale: (1) no importance, (2) minimal importance, (3) average importance, (4) high importance, and (5) extreme importance. Vocational teachers can consider these ratings as benchmarks as to how their peers view microcomputers in vocational teaching.

Instructional units have been packaged in this competency-based resource guide on microcomputer applications for vocational teachers. This handbook is being disseminated by the Curriculum Publications Clearinghouse, Western Illinois University, Macomb, IL 61455.

## UTILIZING A "PROFESSIONAL DEVELOPMENT PLAN" TO INTEGRATE MICROCOMPUTERS INTO VOCATIONAL CURRICULA AND INSTRUCTIONAL STRATEGIES

Vocational educators can carefully examine Category A in the profile (Table 1) and begin to envision how the content of the eight competencies included in this category will enable them to develop a personal plan for microcomputer competency (Tesolowski, Wallin, Roth, & Rankin, 1984). Competency A.1 defines the elements and planning strategies involved in developing a comprehensive plan for implementing computer-based instruction (CBI) in a local education agency (LEA). This instructional unit presents practices that have been implemented in select exemplary programs in the nation. Competency A.2 explores the vocational instructor's role in the plan identified for implementing CBI in the LEA (A.1). Varying practices are reviewed in Unit A.2, which will assist vocational teachers in preparing a microcomputer implementation plan.

The content included in Competency A.3 enables vocational educators to assess their personal levels of microcomputer competency. Self-report test items are included for a representative set of pertinent content areas or domains related to computer literacy. Vocational teachers can identify their strengths and weaknesses on the basis of this self-assessment measure. Upon completing this diagnostic-prescriptive instrument, vocational educators can profile their results on a chart. On the basis of their strengths and weaknesses, vocational instructors can set initial personal goals (Competency A.4) for microcomputer competency.

Competency A.5 facilitates the development of a personal plan for microcomputer competency. Vocational teachers who participate in this unit of instruction are encouraged to develop a Professional Development Plan that includes long-range goals; short-term objectives; and the identification of instructional strategies, methods, techniques, materials, and resources that will facilitate the accomplishment of these goals and objectives. In addition, participants will monitor their timeline in regards to when they initiate and conclude selected learning activities. Finally, vocational teachers will record whether or not they believe they have successfully achieved their goals and objectives. Competencies A.6, A.7, and A.8 assist vocational educators in working through the processes of implementing, evaluating, and modifying their personal plans for microcomputer competency.

After vocational instructors construct their Professional Development Plans for microcomputer competency (A.5), they can implement their plans by fully utilizing all of the units of instruction for the 39 competencies clustered in Categories B, C, D, and E. An alternative to using all of the units of instruction is to selectively choose units based on the needs identified in the personal plans, their districts' or schools' needs, and their personal interests (Roth, Tesolowski, Rankin, & Blackman, 1984).

## THE NEED FOR A PERSONAL COMMITMENT TO APPLY MICROCOMPUTERS IN VOCATIONAL EDUCATION

Competencies identified for this handbook can serve as an invaluable starting point for vocational instructors who want to integrate microcomputers into their professional future. Vocational educators can visually inspect the categories and respective competencies, examine their own teaching situations, and begin to formulate their own individualized plans for applying microcomputers in their programs as well as in their personal lives.

The stage is now set for vocational educators to decide where and how microcomputers will fit into their teaching futures. Competencies identified through this research project can enhance their perspectives of the potential of microcomputers in vocational education. However, vocational teachers must individually develop personal plans for microcomputer competency that will serve their professional needs as well as the needs of their respective programs.

The decision to develop a plan or not is of utmost importance. Plans can be modified as teachers' computing interests and programmatic needs change with the times. Whatever vocational educators personally decide to do, they should not allow this contemporary technology to pass them by. All vocational teachers must critically examine the role of microcomputers in their professional lives.

## REFERENCES

Adams, R. E. (1975). *DACUM approach to curriculum, learning, and evaluation in occupational training. A Nova Scotia Newstart Report*. Ottawa, Canada: Department of Regional Economic Expansion.

Miller-Beach, A. (1980). DACUM: Identifying competencies. *School Shop*, *39* (8), 63.

Pratscher, S. K. (1983, April/May). Computers in schools: A Texas perspective. *Commodore: The Microcomputer Magazine*, pp. 53-55.

Roth, G. L., & Tesolowski, D. G. (in press). The microcomputer freeway: Mapping a future for vocational curricula. *The Journal of Epsilon Pi Tau*.

Roth, G. L., Tesolowski, D. G., Rankin, R. A., & Blackman, H. S. (1984). Getting ready for micros. *VocEd*, *59* (3), 30-31.

Tesolowski, D. G., Wallin, C. W., Roth, G. L., & Rankin, R. A. (1984). The dynamic duo: The microcomputer and the vocational special needs teacher, *6* (3), 17-20.

## Unit 1

# Differentiate Among Applications of Computer-Based Instruction

### UNIT OBJECTIVE

Upon completion of this unit, the learner will be able to differentiate among different applications of computer-based instruction

### SPECIFIC OBJECTIVES

Upon completion of this unit, the learner will:

1) Describe how educational computing relates to the concept "time on task" of vocational students in the learning environment.
2) Explain how the limitations of existing software affect educational applications of microcomputers in vocational education.
3) Differentiate (in terms of educational applications) between tutorial software and drill-and-practice software.
4) Discuss how problem-solving software can be integrated into vocational curricula and instruction.
5) Provide examples of how gaming software may be implemented in vocational education.
6) Discuss the use of simulations in vocational curricula and instruction.

# Differentiate Among Applications of Computer-Based Instruction

## BY: DR. ROGER A. RANKIN

Vocational teachers should subscribe to one important belief: microcomputers are effective tools in vocational curricula and instruction. Teachers need to gain insights regarding the value of using microcomputers for instructional purposes. Microcomputers are not mysterious configurations of chips, drives, and screens that solve the problems generated by learning situations. Microcomputers are, however, tools that contribute to improved learning when they are appropriately applied.

The nature of vocational education creates a favorable learning environment for meshing CBI and vocational curricula. The focus of vocational curricula is a set of predetermined competencies that enable students to develop marketable job skills. The competency-based method for vocational curricula permits instructors to identify specific applications of microcomputers in curricula and instruction. The knowledgable teacher working within a well defined curriculum can identify meaningful uses of computer hardware and software. Vocational instruction (especially CBI) can double student achievement found with traditional instructional methods when materials are carefully selected in advance and when lessons are based upon parameters of student performance (McKeachie, 1974).

The advantage of using microcomputers as learning tools revolves around the concept of "time on task." Educational computing affords students ample opportunities to maintain direct contact with vocational curricula. When students expend more time and effort learning, their levels of achievement increase. The concept of time on task is a more accurate predictor of achievement than many other variables. Microcomputers are very effective for creating high levels of attention in students. Educational computing features visual and other sensory feedback as well as capabilities to provide immediate responses to students. Microcomputers can and do demand high levels of engagement from students (Jernstedt, 1983).

As with any teaching method, the use of microcomputers cannot stand alone, inclusive of the total instructional needs of students, and independent from other segments of the educational setting. Vocational teachers play key roles in the application of microcomputers. Instructors should promote the transfer of knowledge students gain from educational computing to other components of vocational curricula.

## PROBLEMS ASSOCIATED WITH EDUCATIONAL SOFTWARE

The history of educational computing indicates that the application of microcomputers in schools has not reached its instructional potential. In fact, educational computing is a relatively new endeavor that continues to evolve and expand at a rapid pace. Yet computer-based instructional materials and methods appear to be having a minimal impact on the problems which plague vocational instructors. There are numerous explanations for this lack of success, but a readily identifiable reason is the deficiency in the quality and range of software for vocational curricula and instruction (Roblyer, 1983).

Perhaps the most obvious problem with existing software is not that it is ineffective, but that it is limited in what it addresses. Tutorials and software series which thoroughly cover select topics are not as common as one-skill modules and lessons (Becker, 1983). Quite frankly, much of the educational software developed has proven to be ineffective in the classroom.

It is possible that instructors who experimented with early software packages became disenchanted with what was available at the time. Vocational teachers are encouraged to keep up-to-date with the software market. The potential for microcomputers to contribute to vocational curricula and instruction is tremendous. A handicap has been and continues to be a limited supply of software for vocational education. However, as more research and development is devoted to the role of CBI in vocational education this problem will recede.

## EDUCATIONAL APPLICATIONS OF MICROCOMPUTERS

Five educational applications of microcomputers are addressed in this unit:

1. Drill and practice
2. Tutorial
3. Problem solving
4. Simulation
5. Gaming

### Drill and Practice

Historically, most CBI activities have been drill and practice or question and answer tutorials. These software designs continue to be major modes for educational computing (Zinn, 1978). As the name implies, this type of software provides drill or practice exercises for learning facts or principles. Drill-and-practice software functions best as a supplement to other forms of vocational instruction. These software packages permit students to master or refine vocational education concepts or principles at a microcomputer work station. Teachers who use drill-and-practice programs assume that learners possess a certain degree of knowledge and that this type of software will reinforce that knowledge.

Drill-and-practice programs have considerable merit for learning mathematical concepts in vocational education. For example, after receiving instruction in trigonometry functions, a machine shop student may use a drill-and-practice program at a microcomputer station to reinforce this recently acquired knowledge. The program can branch to a prescribed level of mastery of select trigonometry concepts. The machine shop instructor can verify the student's attainment by a quick check of the response rate (Blum, 1982). Students can use the drill-and-practice programs prior to applying trigonometry in the machine shop laboratory.

### Tutorial

Tutorial programs are designed to teach students in similar fashions as instructors. This type of software attempts to simulate a tutor as it introduces lessons, explains, provides hints and examples, asks questions, and evaluates student responses. Additionally, tutorial software gives reinforcement and feedback, and chooses the proper placements for students based on their achievement levels (Hudson, 1980).

Quality tutorial packages demonstrate goal oriented and carefully sequenced patterns. Tutorials are capable of presenting questions, analyzing student answers for patterns of error, and providing remedial instruction to eliminate those errors (Rodrigues, 1984). It is not necessary to augment tutorial software with other types of instruction. Tutorial programs function independently within vocational curricula and do not supplement classroom teaching as do drill-and-practice programs.

Vocational instructors may incorporate tutorial programs for three specific purposes:

1) To assign vocational students to tutorial programs if they need enrichment or reinforcement,

2) To individualize instruction in order to increase the instructor's role as a facilitator of instruction, and

3) To incorporate group discussion and guidance while the course is ongoing on the terminal (Blum, 1982).

Tutorial software and drill-and-practice programs can serve two useful functions for vocational teachers: a) tutorial programs can be used to teach vocational concepts or principles, and b) drill-and-practice programs can provide students with practice activities to reinforce the initial learning of those concepts and principles.

### Problem Solving

Instructors in vocational education have a keen sense of the work skills that vocational students need prior to entering the world of work. However, a bank of knowledge or a handful of skills does not ensure a student's success in the workforce. Vocational graduates must perform in a larger arena that requires abilities which transcend all aspects of work: The ability to solve problems and the ability to think something through to its best solution. Methods by which vocational educators can improve problem-solving abilities merit discussion. Some authors believe that the mathematical solution to a problem constitutes a small portion of problem-solving in the real sense. Problem-solving involves gaining an accurate understanding of a problem and identifying realistic alternatives that will lead to the most effective solution of that problem.

For vocational students to develop problem-solving skills, they must deploy a type of cognitive strategy. This strategy will depend to a great extent upon previously learned rules and upon a type of intellectual skill that governs the learner's thinking process. Problem-solving involves the gleaning of sets of rules that the student has previously developed and then creating a new set of rules to solve the problem. The student can then apply this new set of rules to future situations which resemble the newly solved problem (Gagne, 1977).

Problem-solving situations occur when instructions that students receive do not include stated solutions and learners are required to develop them. Solutions which a student must develop have a high rate of retention in the student's memory. Once students have struggled to develop higher order rules to solve a problem, they will not be easily forgotten (Gagne, 1977).

Problem-solving activities in CBI are of two kinds. First, there are problems students can solve with the skills they currently possess. Second, there are those problems that are not readily solved. To solve these problems, the students must create a higher order of rules, relying on rote memory, experience, and drill-and-practice exercises the student has previously performed. The importance of good problem-solving skills cannot be underestimated because these skills build the cornerstone of an effective worker.

Computer-based instruction provides the potential to create student insights regarding how problems are solved with components of problem-solving models. Microcomputers can acquaint students with a systematic approach to problem-solving. This approach is significantly different from offering students solutions to textbook problems without identifying the steps that lead to the solution of that problem. Problem-solving activities in educational computing can greatly enhance vocational training.

### Simulation

The capstone for vocational education is in place when vocational graduates enter the workforce and apply learned skills, knowledge, and attitudes. Vocational teachers strive to provide meaningful learning experiences which "pull together" the ideas presented in curricula and instruction. In accounting, typewriting, and office procedures, business education teachers utilize practice sets. Agricultural teachers utilize student projects. Trade and industry teachers require that students construct something. Many students develop and refine vocational skills through contests in student organizations and through opportunities with cooperative education programs. On a new learning horizon, simulated activities with educational computing provides students with additional opportunities for learning vocational skills, knowledge, and attitudes.

Computer simulations can be valuable learning activities for students who are developing vocational skills. Simulations help students study situations which they would normally be unable to examine because of danger, expense, or length of time. Computer simulations make abstract subjects more concrete, promote learner inquiry, and spark interest in learning (Roberts, 1983).

Vocational instructors can integrate computer simulations into vocational curricula and instruction where they recognize patterns for problem-solving activities. Simulation software can be infused into the classroom or laboratory setting to simulate a real situation. It can dramatize the components of a problem in detail, introduce randomly selected values and events, and predict the results of a particular series of decisions. Simulations can greatly extend and broaden the learning experience of vocational students (Hudson, 1980).

### Gaming

Anyone associated with a laboratory of microcomputers, or anyone with a microcomputer in the home, can attest to the fact that playing microcomputer games is extremely popular. Many teachers and administrators are quick to eliminate gaming from computer laboratories. This policy may be at the expense of valuable educational experience for students.

Gaming can be a useful component of an educational program if it provides types of experiences similar to those gained in problem-solving and simulations (Stowbridge & Kugel, 1983). While the exact educational consequences of gaming are difficult to identify, there appear to be intangibles that benefit vocational students. Regardless of the game, students learn strategies for improving their success. These games provide ample pressure for students to win, and yet they do not create excessive worry over failure, as do certain testing situations (Seidner, 1976). Games contribute to the improvement of memory and recall, and require students to think about outcomes before initiating actions.

Additionally, gaming encourages students to formulate more than one approach to problem-solving. Gaming activities challenge students by suggesting that an unsuccessful solution for one problem may be applicable in the next similar situation (Stowbridge & Kugel, 1983). With the right approach, the benefits of games can be combined with educational needs to provide learning alternatives for vocational students.

## SUMMARY

This unit explains the educational applications of computer-based instruction. Five applications are discussed:

1. Drill and practice
2. Tutorial
3. Problem solving
4. Simulation
5. Gaming

Drill-and-practice programs require students to have a previous knowledge of fundamentals, rules, or concepts. Drill-and-practice software reinforces the competency levels of users within a given subject.

Tutorial software teaches users the fundamentals, rules, or concepts of a subject without the requirement that students already have knowledge of the topic. Tutorial programs actually work as tutors (or teachers), introducing the subject and providing opportunities for students to initially learn the subject.

Problem-solving programs provide opportunities for students to apply fundamentals, rules, or concepts to situations and to gain additional insights regarding the subject. Problem-solving is not restricted to the mathematical solution of a problem. It includes a greater understanding of a problem and the realistic alternatives that lead to an effective solution of that problem.

Simulations can be the capstone to training in vocational education. Simulation activities require students to apply learned skills, knowledge, and concepts. Simulations provide meaningful exercises which involve students in analysis and synthesis of the elements of a vocational discipline.

Gaming is a broad area which provides entertainment via microcomputers. While games are nct specific to vocational curricula, gaming warrants consideration as a learning activity for students on a controlled basis. Gaming provides opportunities for students to learn strategies for success in an atmosphere which is not threatening. In addition, students can learn that a single approach to problem-solving may not be realistic. Gaming may be employed by vocational teachers as a motivational tool with select students.

## ACHIEVEMENT INDICATORS

1) Describe how educational computing relates to the concept "time on task" of vocational students in the learning environment.
2) Explain how the limitations of existing software affect educational vocational education.
3) Differentiate (in terms of educational applications) between tutorial software and drill-and-practice software.
4) Discuss how problem-solving software can be integrated into vocational curricula and instruction.
5) Provide examples of how gaming software may be implemented in vocational education.
6) Discuss the use of simulations in vocational curricula.

## REFERENCES

Becker, H. J. (1983). How schools use microcomputers. *Classroom Computer Learning, 4* (2), 41-44.

Blum, V. L. (1982). Evaluating instructional software for the microcomputer: An analytical evaluation procedure. *Dissertation Abstracts International, 43,* 3158A-3251A (University Microfilms No. 83-45, 57).

Gagne, R. M. (1977). *The Conditions of Learning.* New York: Holt, Rinehart, & Winston.

Hudson, C. J. (1980). Computer assisted instruction in agricultural education. *The Agricultural Education Magazine, 53,* 19-20.

Jernstedt, G. C. (1983, May). Computer enhanced collaborative learning: A new technology for education. *Technological Horizons in Education Journal,* 96-101.

Roberts, N. (1983). Testing the world with simulations. *Classroom Computer News, 3* (3), 28-31.

Roblyer, M. D. (1983). Toward more effective microcomputer courseware through application of systematic instructional design methods. *Association for Educational Data System, 1* (1-2), 23-31.

Rodrigues, R. J. (1984). Computer program models for teacher education. *Action in Teacher Education, V* (4), 15-21.

Seidner, C. J. (1976). Teaching with simulation and games. The psychology of teaching methods. *The Seventy-Fifth Yearbook of the NSSE, Part 1.* Chicago: University of Chicago Press.

Stowbridge, M. D., & Kugel, P. (1983, April). Learning to learn by learning to play. *Creative Computing,* 180-188.

Zinn, K. L. (1978). Sources of information about computing in instruction. *Educational Technology, 18* (4), 29-32.

Zinn, K. L. (1979). Using the microcomputer: One university's experiences. *Educational Technology, 19* (1), 54-55.

## Unit 2

# Assess Students' Needs For Specific Computer-Based Instruction Applications

### UNIT OBJECTIVE

Upon completion of this unit, the learner will be able to make curriculum and media decisions based upon student needs and desired learner behaviors. This knowledge will be demonstrated through completion of the achievement indicators at the end of this unit.

### SPECIFIC OBJECTIVES

Upon completion of this unit, the learner will be able to:

1) Choose appropriate media (microcomputer software) to match student needs.
2) Organize curriculum material according to appropriate CBI tasks for learners.
3) Organize instruction into learning objectives which facilitate media, teaching, and student performance divisions.
4) Develop a preassessment instrument with regard to a unit of instruction which reflects predetermined objectives.
5) Place students in competency settings which reflect the knowledge levels which students bring to learning settings.

# Assess Students' Needs for Specific CBI Applications

### BY: DR. JOHN A. PIEL

One of the most important practical teaching decisions made by vocational instructors is the choice of instructional media. Language media, test media, and CBI media are examples of curriculum and instruction alternatives which teachers encounter. The delivery of instruction language media is often combined with actual objects. Vocational instruction relies heavily on this combination of media because vocational students are often required to develop motor skills which will prepare them for the world of work. It is this combination which permits CBI to be best put to use by vocational teachers. Vocational instruction which closely approximates the real world of work enhances learning (Gagne, 1977). Unit B.2 and C.1 present a myriad of educational applications for microcomputers. As these applications move closer and closer to paralleling computer applications within the workforce, students will significantly surpass the instructional effectiveness of traditional passive models of instruction which rely on teacher centered lecture methods.

For teachers to best decide how or when to incorporate CBI, the requirements of learning tasks must be analyzed. This analysis includes task description, learning analyses, deriving external conditions, and evaluation. These procedures are presented in this unit to help vocational teachers develop strategies for assessing students' needs for CBI applications.

**Task Description**

Students are capable of performing a multitude of tasks from simple tasks, such as opening a door, to complex tasks of intellectual abstraction, such as mastering calculus or trigonometry. Describing a task may be as complex as the task itself. Describing tasks for workers in job settings is a means of specifying the exact roles and responsibilities of employees. It is a process of matching interests, skills, and aptitudes of workers with the performance requirements of jobs within an organization. Vocational teachers would use a similar process to delineate learning tasks for students enrolled in vocational programs. Describing the tasks of a learning process can enhance learning efficiency within a classroom and lead to a smoother and more productive curriculum and instruction process. From the standpoint of learning analysis, expected learning outcomes should be described to students in advance of instruction.

A detailed description of learning tasks will help optimize student performances. First, learning tasks should be categorized according to expected outcomes which can be grouped into one of five categories (Gagne, 1977): a) intellectual skill, b) cognitive strategy, c) information, d) motor skill, or e) attitude. As an example of one of these categories, CBI can assist learners in the development of computer programming skills, which requires students to employ analytical approaches to problem-solving. Students engage in a definite and detailed cognitive strategy or information processing as they develop a computer program which commands a computer to perform an assignment. As students become proficient at educational computing and interact more with CBI programs, they will become less dependent on instructors as sole sources of information. Microcomputers will then serve as information processing agents with students working as the consumers of that information.

A distinct set of motor skill requirements must be developed in order for students to be proficient users of CBI materials and methods. Keyboarding skills will probably be the most important set of skills for vocational students to develop. Students can still be computer users without keyboards skills, but their proficiency will be significantly diminished.

### Task Description as Learning Objective

The purpose for using learning objectives in a computer-based instruction program is to specify those student performances which are appropriate for the select content of vocational curricula. This process solidifies the relationship of Task Description to learning objectives. Using the Task Category of Intellectual Skill as an example, a behavioral objective is not only valuable in planning for student performance, but, when correctly written, can be beneficial in charting teacher performances and evaluating the appropriateness of selected media for a given unit of instruction. The following narrative is intended to aid vocational teachers in comprehending the relationship between the learning objectives of a CBI program and the educational effectiveness of teachers and media in the vocational education setting.

### Writing Learning Objectives

Learning objectives include three distinct components which relate to the instructional effectiveness of the teacher: a) media selection, b) implied or described teacher behaviors, and c) measurable student performances relative to a predetermined criteria of behavior. The following statement is an example of an objective which pertains to the Learning Category **intellectual skill knowledge** for the vocational education discipline of office occupations:

Upon completion of the IBM Assistant Series Software package and a discussion regarding implications for today's business community, the student will be able to match a specific office job with an assigned work task.

The three components of the preceding learning objective are easy to identify. The media component is identified by the statement stem. . . "completion of the IBM Assistant Series Software package." This stem implies that teachers w ill go beyond the lecture method and include media which more closely replicate an automated office. Not only will students become involved with technology presently employed in many offices, but valuable initial bits of vocational information can be represented and developed in an individualized manner.

The second component of the learning objective is an implied or described teacher behavior. This is inferred by the phrase . . . "discussion regarding implications for today's business community." This phrase implies that the completion of the computer software package is not a sufficient development of intellectual skills.

A final concern for instructors when writing instructional objectives is to ensure that traditional teacher behaviors are defined. After students have developed **base line** skills through completing the software package, this information must be related to a job-specific application. In other words, vocational instructors must relate general information of the software program or its documentation to specific job applications through discussion, lecture, or hands-on activities. This third component of a properly written learning objective can be identified in the preceding example by distinguishing student performance. This performance is specified by the statement . . . "the student will be able to match a specific office job with an assigned work task." This match may take place by completing a written test, by working through a learning activity packet, by participating in a group discussion, or by interfacing with another software package. The assessment strategy is usually selected by the instructor.

All curriculum planning for the educational computing segments of a vocational program can be organized around instructional objectives regardless of the learning tasks. A prerequisite task for vocational teachers as they develop CBI materials and methods is to write and use learning objectives which can be precisely and distinctly measured. Regardless of the level of the learning task, this precise measurement is required for instructors to assess the level of student performance. Measurable objectives permit teachers to assess the necessity for follow-up activities with supplemental materials or media packages that will help students meet predetermined curricular objectives.

The work of Gagne and Briggs (1974) can help vocational instructors examine the capabilities of students and associate those skills with learning tasks. Learning tasks can then be incorporated into the learning objectives of a CBI program. The following chart presents human capabilities, verbs which depict those capabilities and examples of phrases which may be used to form learning objectives:

| CAPABILITY | VERB | EXAMPLE |
|---|---|---|
| Intellectual Skill Discrimination | DISCRIMINATES | Discriminates, by matching hard lead pencils with desired line widths for a mechanical drafting project. |
| Concrete Concept | IDENTIFIES | Identifies, by responding to a graphic display on a microcomputer monitor, the lubrication points of a truck chasis. |
| Defined Concept | CLASSIFIES | Classifies, by using predetermined characteristics of metals as ferrous or non-ferrous. |
| Rule | DEMONSTRATES | Demonstrates, by solving problems presented through a software package, the ability to read a micrometer. |
| Higher-Order Rule (Problem Solving) | GENERATES | Generates, by synthesizing applicable rules, a paragraph describing a dental assistant's reaction to a patient exhibiting fear. |
| Cognitive Strategy | ORIGINATE | Originates a solution to reduce a customer's hostility by applying learned concepts for effective interpersonal communication skills. |
| Information | STATES | States orally procedures for installing wiring for a specific type of low voltage control. |
| Motor Skill | REPLACE | Replaces a thermocouple. |
| Attitude | CHOOSES | Chooses technician's apparel which portrays a neat and professional appearance. |

The preceding examples contain verbs which describe student performances in measurable terms. Verbs such as "appreciate," "understand," or "know" should be avoided by vocational instructors due to their lack of precision when it is time to determine satisfactory levels of performance. It is very difficult to measure how much a student **appreciates** the application of a specific learning task to a specific job performance.

Similar procedures for planning and implementing instruction may be followed for the other four instructional task areas. Vocational teachers should follow the behavioral objective format using the three component model. In addition to the three component areas, teachers must continually scrutinize the verbs which they select for the learner objectives of educational computing instructional packages.

### Examining Learning Tasks With Regard to Prerequisites

When learning is to take place, vocational teachers should have an overall concern for what learners bring to respective learning situations. In short, instructors should know what learners presently know regarding what is to be taught. Glaser (1967) discusses entering criteria for successful behavior to occur. To gain necessary information, teachers may use pretesting.

This concern may best be addressed with an example. Student A enters an office occupations program with work experience as a secretary and an understanding of how an office should be organized and managed. Student B enters the same class without prior knowledge or experience concerning office management. A preassessment of each student's knowledge and skills can help teachers place them in logical levels within a vocational program.

Once students have been preassessed, two distinct programs will appear: a) a program for those learners who have essential prerequisite capabilities, and b) an additional program organized to branch prerequisites in a sequential fashion. In both cases, curriculum and media discussions should be based upon a learning task analysis and a classification of the five identified learning task headings. Preassessment material should be developed from the formation of questions developed directly from instructional objectives. It is part of the role and responsibilities of the vocational instructor to preassess the knowledge and skills which students bring to a vocational program. By developing appropriate preassessment techniques, needless relearning of old material can be minimized and the appropriate placement of students within the program becomes much more manageable.

## SUMMARY

Computer-based instruction is a new wave crashing the gates of vocational education. A great deal of the features of CBI—the jargon, hardware, materials, and methods—are unfamiliar to vocational educators. However, instructors should not disregard the concepts which have been used for many years to formulate quality vocational curricula and instruction. Teachers will be able to employ those traditional approaches to developing curriculum and instruction in assessing student needs for specific CBI applications. Microcomputers are tools for educational applications—decisions for their usage should be based upon instructional objectives which delineate media, teacher, and student performance requirements.

## ACHIEVEMENT INDICATORS

1) Choose appropriate media (microcomputer software) to match student needs.
2) Organize curriculum material according to appropriate CBI tasks for learners.
3) Organize instruction into learning objectives which facilitate media, teacher, and student performance divisions.
4) Develop a preassessment instrument with regard to a unit of instruction which reflects predetermined objectives.
5) Place students in competency settings which reflect knowledge levels which students bring to learning settings.

## REFERENCES

Gagne, R. M. (1977). *The conditions of learning*. New York: Holt, Rinehart, & Winston.

Gagne, R. M., & Briggs, L. J. (1974). *Principles of instructional design*. New York: Holt, Rinehart, & Winston.

Glaser, R. (1967). Some implications of previous work on learning and individual differences. In R. M. Gagne (Ed.), *Learning and individual differences*. Columbus, OH: Merrill.

## Unit 3

# Developing Lesson Plans For Incorporating Computer-Based Instruction

## UNIT OBJECTIVE

Upon completion of this unit, the learner will be able to develop lesson plans to incorporate CBI into a vocational program. This knowledge will be demonstrated by the development of appropriate lesson plans for the classes taught by the instructor and through completion of the achievement indicators at the end of this unit.

## SPECIFIC OBJECTIVES

Upon completion of this unit, the learner will be able to:

1) Discuss the four stages of the teaching-learning process.
2) Organize components of a lesson plan into proper sequence.
3) Identify various instructional methods which can be used to incorporate CBI into the vocational program.
4) Identify various types of software which can be used to teach CBI concepts to students.
5) Select appropriate software to use with various methods of instruction.
6) Develop lesson plans which could be used to teach:
   a. introduction to computers
   b. keyboarding skills
   c. drill-and-practice skills
   d. financial applications
   e. communicating skills
   f. simulation activities
   g. task performance skills

# Developing Lesson Plans For Incorporating CBI

### BY: DONALD ESHELBY

The teaching-learning process consists of four distinct parts which comprise a successful lesson plan:

1. Preparation
2. Presentation
3. Application
4. Evaluation

1. The preparation stage includes tasks such as: review the objectives and unit content, read suggested activities, and select and disseminate student material.
2. The presentation or delivery stage includes tasks such as: provide activities for motivation, use appropriate teaching strategies, demonstrate procedures, and explain safety rules and precautions.
3. The application stage takes place during the time students practice skills, participate in learning activities, complete job or assignment sheets, or perform other activities as provided in the objectives.
4. The evaluation stage takes place during the time students take tests and demonstrate motor skills, and teachers document proficiency or note progress. The teacher reteaches if and when necessary.

These four stages are generally known as the Four Step Lesson Plan or Four Step Method of Teaching.

## COMPONENTS OF A LESSON PLAN

A typical lesson plan consists of a list of items an instructor intends to teach when setting up a classroom or laboratory activity for students. Every aspect of the activity should be identified. A lesson plan is generally prepared for one period of instruction, but may be designed for a series of related activities which cannot be completed in a single class period. A lesson plan differs from a Unit of Instruction in that a Unit of Instruction provides material for an entire group of related assignments or tasks. There may be many lesson plans for one Unit of Instruction, depending on the topic to be covered and the depth of instruction needed.

*The components of a typical lesson plan include:*

1. Topic
2. Objective(s) to be covered
3. Material, supplies, or equipment needed
4. Assignment (usually given beforehand)
5. Introductory information
6. Presentation or demonstration
7. Applications or student performance desired
8. Evaluation
9. Time elements

Lesson plans are designed to provide teachers with road maps of activities which are essential to providing meaningful learning experiences for students. Some instructional packets include a suggested activities page which includes references to source material. This provides the instructor with material from which to assign reading activities for in-depth learning or for remediation.

It is an excellent idea to include a checklist of tools, supplies, equipment, and visual aids needed for the lesson. Most instructional packets provide job sheets on which these items are listed for the student. However, an instructor's needs are generally more substantial than a student's. Instructors should review this list and test all equipment before class begins.

Content specification and content analysis are best done before classroom activity begins. Content analysis is divided into two blocks: concepts and tasks. The teaching of tasks is generally easier than the teaching of concepts. Tasks are best taught by demonstration. Concepts may be taught through class discussion and visual presentations. If different teaching methods are involved, they must be noted in the lesson plan.

During the delivery stages of the lesson, instructors should be flexible and thorough in covering topics. Individual learning styles can be accommodated by adjusting lesson plans to provide the necessary review of difficult topics or to establish large and/or small group instruction.

## INSTRUCTIONAL METHODS OR STRATEGIES

An array of instructional strategies can be used to present classroom material using CBI formats. The method to be used should be consistent with the content of the topics being discussed and should complement the learning styles of the students. CBI is an excellent way to motivate nonachievers or underachievers. Sample teaching methods which can be used include:

1. Small group activities
2. Peer teaching
3. Simulations
4. Gaming
5. Drill and practice
6. Dialogue
7. Individualized instruction

Small and large group activities for CBI do not differ significantly from traditional activities except that some form of microprocessor is utilized in the classroom rather than textbooks or lectures. The lecture is still the most effective method of providing general information to a large group, but it should be used sparingly. The general pattern to establish is to rotate two or three students using one piece of equipment to ensure that each student is able to use the equipment.

Peer teaching is particularly attractive in CBI because, quite often, students have more knowledge about computers than a beginning CBI teacher. Because some students acquire computer skills very quickly, it is more efficient to allow these students to share their abilities with other students. This further reinforces their learned skills and provides a teacher's aide for the classroom.

Simulations are effective methods of teaching when the material or topic being covered is hazardous, supplies are expensive, or sensitive equipment must be used for final results. For example, a chemistry experiment can be performed by simulation using the appropriate software packages. This approach helps to prevent costly accidents in the classroom. Testing a piece of electronic equipment which is both expensive and scarce can, likewise, be conducted under simulated conditions. Experience or encounter simulations, such as a stock market activity, foreign country travel, or becoming a parent, can be implemented in a vocational program.

Other simulations can be set up to prepare a student to drive, operate equipment, construct a building, make a medical judgement, alter an ecosystem, or explore human anatomy. These surrogate experiences can prepare students for hands-on practice without costly supplies being wasted. Task performance simulations, such as driving or construction, are goal oriented. Systems simulations are usually process oriented. Encounter or experience simulations can be set up to provide opportunities for students to internalize feelings or to expand ideas and concepts.

Gaming activities are excellent motivational devices for all types of students, from the slow learner to the gifted. A common stock market game allows students to make a million dollars or lose their resources without consequence. Simple exercises can be used for word usage, spelling, mathematics, and social skills.

There are two types of games: instructional and recreational. Both are useful and can teach competitiveness, personal achievement, and individuality. Instructional games with content closely related to the objectives of the lesson can be more rewarding than conventional classroom motivators. Many games also require cooperation from two or more students to achieve the objective of the game. In particular, games can teach problem-solving skills.

Drill and practice, a common teaching strategy, is well suited to the traits of the computer. A computer never tires of repeated exercises and will always provide encouragement if it is programmed to do so. Paired associate materials (such as those explored by Skinner in stimulus/response situations) are related to drill-and-practice exercises. For example, learning French/English vocabulary or technical terms are applications of this type of teaching technique. Drill-and-practice exercises can provide material from stored lists of items, or they can generate materials according to a formula or a set pattern.

Dialogue approaches to CBI are probably the most common applications of computer assisted instruction. These programs are designed to emulate the dialogue between a teacher and a learner. Interaction is required to obtain the desired information. There are two types of dialogue used in CBI: tutorial and inquiries. Tutorials are computer-controlled, while inquiries are learner-controlled. Tutorial methods generally give the answers to students. Inquiry packages generally require learners to extract information through a series of responses and alternative actions.

Individualized instruction is another method of utilizing CBI in the classroom. It is an efficient way to provide remedial instruction to students who need additional assistance or to provide students who have mastered concepts an opportunity to advance to other topics. The early use of computer assisted instruction involved individualized instructional methods which were, for the most part, programmed learning or page turning exercises without dialogue. Poor teaching practices resulted in little or no learning taking place. In fact, interaction with the computer and instructor is critical to individualized instruction. The learner may progress at a rate consistent with his or her abilities, but the whole process is dependent on the sound educational principles of the student/teacher relationship.

### Types of Software to Teach CBI

Five major types of software are in general use in industry and schools (for further information, see Unit C.1):

1. Tutorial
2. Spreadsheet
3. Simulation
4. Application
5. Data Base Management

All five software types can be used in vocational programs. The simplest type is tutorial software, which can be either very elementary or very complex. It is most often used at the beginning level. Drill and practice is a common application of this software. The more effective packages allow for learner interaction and both correct and incorrect responses by the user. The keyboarding skills needed by students can be effectively taught using this software. A step-by-step procedure will teach even the most complex financial or word processing concepts to novices if the package is properly developed by the software publisher. Most software packages are equipped to handle a multitude of educational levels in a single application.

Spreadsheet software is very popular with financial or accounting teachers because of the versatility of the material. The software can handle hundreds of rows and columns of data, both verbal and numerical, and can provide equation results, along with sorting and filing functions. Instructors can utilize these software packages to produce reports as well as charts and graphs. As a rule, the more capabilities they have, the more expensive they are. These integrated packages approach the level of database management packages, although they don't have the capacity to handle as much data and they don't process those data as rapidly. They are best used for accounting applications.

Simulation software packages are very diverse in their applications to educational fields. They can range from a simple game to psychometric application. A number of advanced packages are utilized by industry to train personnel managers and salesforce people to deal with problem situations. There are three basic types of simulation materials: a) task performance, b) systems models, and c) experience or encounter types.

The task performance types are designed to assist a person to learn a specific task or series of tasks. They are generally set up to increase the student's level of learning through continued practice of the tasks. Models of systems are used to simulate a real situation in a controlled environment. Students are provided an opportunity to manipulate various parameters of the system and examine the results of those changes. They can learn to understand the operations and processes involved or they can grasp the concept as a whole after they have completed the exercise.

Experience or encounter simulation packages allow for the unknown to enter the classroom through such applications as family relationships, financial forecasting, or for something as simple as clothes choice. Industry applications can be made to manufacturing schedules and cost studies. Diagnosis in the mechanical fields is a natural application of this type of software package, as are drafting and design applications. The "what if" application is a powerful teaching tool.

Application type software are generally industry generated software for specific tasks within the industry (such as computer controlled machinery). Some word processing software is directly applicable to industry. If, for example, the banking industry utilized a specific brand of software in all of its offices, it would be wise for schools to prepare students with an understanding of that particular software. Very few types of software do not have some application to industrial settings. For example, tutorial applications are often used in industry to familiarize new employees with equipment. However, the applications type of software is generally associated with those specific software packages which have been tailored for a particular task using a specific machine to perform a specific function.

Communications software is another application type common in industrial settings. This software establishes a communication link between different computer related equipment. It enables equipment to communicate with other pieces of equipment, or allows two or more computers to exchange information. Database files located in a central office can be assessed by salespersons through a communications network using telephone lines and portable terminals at remote sites. Inventory can be checked routinely and replacement stock ordered as needed using this process.

Database management software, widely used in several fields, can handle many sets of data and can file, retrieve, sort, print, adjust, perform mathematical tasks, and remind people of appointments. The most common application of this type of software is in the planning and project tracking or estimation areas. Instructors can use these packages for management of classroom activity, grades, student projects, competency achievement by students, and inventory of supplies and materials used on a daily basis. Most of the computer-controlled machine software in industry is a form of database management software. Attorneys can establish pleas for their clients using information retrieved from law records in previous trials. There are unlimited applications for databased management software. The major drawback to the package is the high cost which is directly related to the quality of data which can be handled. Most integrated software packages have a strong database function as the core component.

**Selection of Appropriate Software**

The number of software packages available is staggering. The choice is up to the individual instructor. One should review the material before purchasing it if possible. The next best thing to reviewing it yourself is to obtain a report completed by a renowned critic in the field. The Northwest Regional Education Laboratory, which reviews software packages, publishes reviews in an electronic database, RICE (Resources in Computer Education), and provides hardcopy on request. Most computer magazines feature software reviews in a monthly column.

The critical point to remember is to purchase software designs to provide the needed instructional information. Instructors should not attempt to prepare courses on the basis of available software. Teachers should focus on the content of the vocational program. The software for CBI is simply a tool for use in the classroom. You don't buy a four wheel drive tractor and then decide to purchase a farm.

The examples which follow in this section are for references only. There are hundreds of examples to use, but these were chosen as simple applications for the methods of instruction identified in a preceding section of this unit.

A simulation package in the health field, which is based on the struggle between the body's immunization system and viruses, is titled CELL DEFENSE. The package is more of a game than a true simulation, but it is entrancing to students who learn the body's natural defense mechanisms by using the software. The manual accompanying the package does an adequate job of explaining cell structure. A beginning level software, it is an easy package to learn.

Another simulation or decision-making package, titled CHIEF EXECUTIVE SERIES, contains educational business management games designed to teach the fundamentals of decision-making and strategy through role-playing situations. Designed for more advanced instructional levels, it is suitable for Marketing and Business Education classes. Any program which provides entrepreneurial skill could be applicable to business students.

KNOWARE AT HOME is a software package designed to prepare novices with an understanding of word processing, database management, spreadsheet, and financial decision-making. It is especially written for those students or home users who are not yet familiar with their computers. It is a relatively easy package to use. For Apple users who are quite well versed in word processing using APPLEWRITER, there is MAESTRO. This package will allow the student to become proficient in word processing skills. Another typing package which is a tutorial design rather than word processing is called MASTERTYPE. This package teaches the Dvorak keyboard skills using Apple equipment.

Many software packages have been developed for the instructor's use. This type of materials include packages such as TESTMASTER, which will allow the teacher to develop test questions: multiple choice, true-false, completion, and short answer type questions. Multiple choice, for instance, can be generated by the teacher and managed in separate files. Many of these types of software are available through computer user groups developed by frustrated instructors who were searching for the perfect test package not available in commercially developed software.

Drill-and-practice material is plentiful because this was the stock-in-trade for software developers for many years. Properly selected, drill-and-practice materials can be very useful to the classroom instructor. THWART! is such a package, written to develop vocabulary skills for all levels, from children to adults. It uses a crossword puzzle approach. SPELLING BEE is another drill-and-practice set for all ages. WRITING SKILLS is a package designed to provide the learner with skill in Language Arts by explaining comma, pronoun, modifiers, and other word usage.

THE OREGON TRAIL is an excellent decision-making software package used by many computer stores to show the power of the computer as a game and educational tool. This package leads students through a covered wagon expedition across early nineteenth century America. The need to stock food, ammunition, clothing, and personal items is stressed. Enroute, the user's decisions force the expedition to stray from its chosen path when detours and hardships from the elements and hostile groups are met. This software is an example of the experience simulation models available to teachers.

Many excellent financial, accounting, or spreadsheet software packages are available. The industry-related models include LOTUS 1*2*3*, MULTIPLAN, VISICALC, and PEACHCALC. VISICALC, a very popular package, is the benchmark software which made microcomputers popular in businesses. These and many others are applicable in an Office Education program as applied to accounting or bookkeeping courses. They are actual working models, not games, and can be used for making projections, balance sheets, and other industrial uses. They are a direct application of CBI because the student must learn to master the computer and the software package.

Industrial applications of word processing software include WORDSTAR, PERFECT WRITER, PEACHTEXT 5000, WORDPERFECT, and many more. Unlike the spreadsheet software previously described, only a few of these software packages have been accepted by industry. The wise vocational teacher will survey industries in the community to determine which software is being used.

The type of software which has direct industry application is task development oriented when used in an educational environment. The teaching method is performance directed and should be evaluated in an appropriate manner. For example, letter perfect or mailable correspondence is a true industry standard, as are error free account ledgers. The student should perform as an industry employee after appropriate learning or practice interval has been provided.

Programming computers to be used in a manufacturing situation is another CBI application. The software for this educational experience is generally provided by the equipment manufacturer. Many models are available to teach industry simulation, and actual CAD (computer assisted design/drafting) or CAM (computer assisted manufacturing) software packages can be purchased from commercial sources. These are generally expensive, as is the equipment needed for the application.

The major points to consider when deciding which application to use revolve around the classroom objectives and the industrial setting in which the students will be seeking employment. Cost and equipment needed to complete the exercise are other considerations.

## Lesson Plan Development

Each lesson should consist of the necessary information a teacher needs to prepare and conduct courses in the classroom, laboratory, or other setting. The lesson plan is prepared for each teacher usage. It is a guide to the structure and sequence of the topic within a specified timeframe. Nine components should be included in each plan.

1. Topic
2. Objectives
3. Material, equipment, and supplies
4. Assignments given
5. Introductory information
6. Presentation or demonstration
7. Desired student performances (application)
8. Evaluation
9. Timeframe (length of lesson)

The components do not have to appear in the order suggested, but each one needs to be described so that no element is overlooked (such as preparing overhead transparencies, or trying to use a faulty piece of equipment). Sample lesson plans are provided for various methods of instruction or CBI applications.

### LESSON ONE

TOPIC:
Introduction to Computers

OBJECTIVE(S):
a) Introduce students to the computer.
b) Describe uses of the computer.
c) Identify major parts of the computer.
d) Demonstrate the operation of the computer.

MATERIAL, SUPPLIES,
AND EQUIPMENT:
Introduction to Computers handout; computers and peripherals; printer paper; sample programs; list of terminology; transparencies (intro 1-7); spare computer parts; and circuit cards.

ASSIGNMENT:
Review the reading assignment and discuss terminology.

INTRODUCTORY
INFORMATION:
Discuss how computers and people differ; how computers handle information; why computers are important in the lifestyles of today.

PRESENTATION/
DEMONSTRATION:
Show overheads on the evolution of computers; explain the parts of a microcomputer using examples from old machines; demonstrate how a computer is operated—use Hangman program.

PERFORMANCE/
APPLICATION:
Lecture about and demonstrate computers; show students how to turn on and operate a simple program using written directions.

EVALUATION:
Students judged on completion of assignment; answers to oral questions; ability to follow directions and operate the equipment.

TIMEFRAME:
Two class periods, second one for practice and review of information.

## LESSON TWO

TOPIC:
Keyboarding Skills

OBJECTIVE(S):
a) Discuss keyboarding and its importance to personal needs in the future.
b) Describe and demonstrate the touch method of keyboarding skills and explain the different styles of keyboarding.
c) Introduce students to keyboarding numeric and alphabetic information by touch.

MATERIAL, SUPPLIES, AND EQUIPMENT:
Handouts on keyboarding; computers and peripheral equipment; printer paper; text; Keyboarding for Information Processing, Gregg; transparencies (8, 11-15); word and number lists; Scripsit programs.

ASSIGNMENT:
Review the reading assignment and discuss terminology.

INTRODUCTORY INFORMATION:
Describe the differences between typing and keyboarding; present reasons for the need for keyboarding; explain differences between the computer keyboard and the typewriter with the related equipment.

PRESENTATION/ DEMONSTRATION:
Show overheads on keyboard styles; demonstrate the processes used by a computer to store and retrieve data; demonstrate the touch method of keyboarding.

PERFORMANCE/ APPLICATION:
After presentation of information and demonstration of touch method, the students will be expected to practice touch entry for numeric and alphabetic data.

EVALUATION:
Students judged on completion of assignment; answers to oral questions; ability to follow directions and operate the equipment.

TIMEFRAME:
Two class periods, second one for practice and review of information. Continued practice for speed and accuracy will be accomplished over a period of two weeks using advanced work lists and 9 column numbers.

## LESSON THREE

TOPIC:
Drill-and-Practice Skills

OBJECTIVE(S):
a) Improve spelling skills.
b) Provide positive learning environment through a competitive activity.
c) Introduce students to automechanics terminology.

MATERIAL, SUPPLIES, AND EQUIPMENT:
Spelling Bee software; IBM computers; additional word lists on software (introductory level automechanics).

ASSIGNMENT:
Review mechanics terminology from Crouse text.

INTRODUCTORY
INFORMATION:
Outline the rules of the contest and establish teams.

PRESENTATION/
DEMONSTRATION:
Demonstrate operation of the software. Select teams by numeric assignment.

PERFORMANCE/
APPLICATION:
Students inquire about words not clear after the reading assignment. Contest will determine spelling abilities of individual members of the teams.

EVALUATION:
Students judged on performance in contest and manner of support of team.

TIMEFRAME:
One class period, depending on successful review of words at the beginning of class. May be repeated in other classes as a motivation tool.

### LESSON FOUR

TOPIC:
Financial Applications

OBJECTIVE(S):
a) Students to describe the procedures for setting up a checkbook balancing activity on the computer.
b) Discussion of essential information regarding the proper method to use a checkbook.
c) Utilize a computer to manage a checkbook record.

MATERIAL, SUPPLIES,
AND EQUIPMENT:
Apple II computers; Checkbook software programs; job sheets for checkbook data entry; printer paper; overheads 23-27.

ASSIGNMENT:
Review checkbook balancing procedures in text, pp. 56-62.

INTRODUCTORY
INFORMATION:
Review procedures for leading the software and basic checkbook procedures.

PRESENTATION/
DEMONSTRATION:
Present review of checkbook processes on overheads; describe the loading procedures for software; provide job sheets.

PERFORMANCE/
APPLICATION:
After presentation of information and review of the procedures, students practice activities.

EVALUATION:
Students judged on completion of assignment; answers to oral questions; and ability to follow directions and operate the equipment.

TIMEFRAME:
One class period. Review set for following week.

### LESSON FIVE

TOPIC:
Communication Skills

OBJECTIVE(S):
a) Students to describe the procedures for setting up a communication software package and necessary equipment.
b) Discuss the equipment and relationship of each piece of equipment to the communications network.
c) Conduct a communications session.

MATERIAL, SUPPLIES,
AND EQUIPMENT:
IBM computers; communications software programs; printer and paper; working telephone hookup; overheads 33-36.

ASSIGNMENT:
Read chapter eight in the text and chapter seven in IBM Asynchronaus Communication manual.

INTRODUCTORY
INFORMATION:
Review the basics of communication networking and describe the procedures for operation of equipment.

PRESENTATION/
DEMONSTRATION:
Show overheads on equipment used in communications; describe the loading procedures for software; provide time to practice; establish teams for communication tag game.

PERFORMANCE/
APPLICATION:
After practice time has been taken, and overview of the communications network completed, teams of three students are selected to send a message to another, set up equipment, load software, send message, and sign off in specified time.

EVALUATION:
Students judged on accuracy of message; proper use of equipment; and speed of completion of task.

TIMEFRAME:
One class period following classes of explanation and demonstration. Repeat games are to be scheduled at the end of classes as appropriate for motivational use.

## LESSON SIX

TOPIC:
Simulation—Management/Decision-Making Skills

OBJECTIVE(S):
a) Provide opportunities for decision-making.
b) Discuss situations needing agreement by others.
c) Compare decisions to be made by various business ventures and reasons for the type of decision.

MATERIAL, SUPPLIES,
AND EQUIPMENT:
Computers; Chief Executive software; printer and paper; handouts on decision-making.

ASSIGNMENT:
Complete chapter 8 questions in text and prepare for class discussion and quiz.

INTRODUCTORY
INFORMATION:
Hand out quiz on decision-making and discuss major areas before starting computer activity.

PRESENTATION/
DEMONSTRATION:
Provide handout on types of businesses and different methods to deal with problems. Discuss cash flow, capital investment, risk taking, and hedging. Describe procedures for using the Chief Executive series. Show examples of responses to the program's questions.

PERFORMANCE/
APPLICATION:
Students will be expected to complete 3 exercises on the computer after the discussion of the various methods used to handle situations in different businesses. Students who obtain 80% or more on the quiz are to participate. The remainder review the information until sufficiently prepared. This is the first period of computer interaction. It will be repeated four times so that all students have used the computer at least three times. All students will participate in each of the three series of business situations.

EVALUATION:
Student progress to be judged on success in completing the program and on responses on quiz. Proof of success will be provided by a signed copy of a printout from the computer.

TIMEFRAME:
One class period followed by three sessions of computer interaction for each student.

### LESSON SEVEN

TOPIC:
Task Performance Skills—Sales Techniques (Level 1)

OBJECTIVE(S):
a) To improve sales skills.
b) To identify client analysis methods.
c) To assess personal sales characteristics.
d) To determine sales strategy based on customer and salesperson characteristics.

MATERIAL, SUPPLIES, AND EQUIPMENT:
Sales Edge software; computers; printer and paper; handout on successful sales strategies; directions for using software package.

ASSIGNMENT:
Read chapter 6, pp. 132-150, and chapter 8, pp. 213-224.

INTRODUCTORY INFORMATION:
Review of previous discussion of successful sales methods based on client assessment. Description of software package components which are to be used.

PRESENTATION/ DEMONSTRATION:
Students will operate the equipment using the assigned software and will rotate through four stations with four students to a station.

PERFORMANCE/ APPLICATION:
Small group discussion will be the main delivery used. Students will be expected to prepare a self assessment of sales characteristics to be used in the group discussion sessions and will prepare sales strategies to compare with the computer's analyses. Student results will be compared to the analyses provided by the computer software.

EVALUATION:
Students will be judged on performance using the computer analyses to compare sales styles and on discussions of reasons for differences. Computer printouts of student's self assessment will be retained for future comparison to determine growth and understanding of sales techniques.

TIMEFRAME:
Four class periods, followed by a repeat activity every two weeks to reinforce knowledge. This is level one in the software application. The package will be used in most classes on a periodic basis.

## ACHIEVEMENT INDICATORS

1) Discuss the four stages of the teaching-learning process by giving examples of each as used in a class lesson.
2) List the components of a lesson plan in sequential order as you would for preparing to teach a class.
3) Identify four instructional methods to be used when incorporating CBI in the vocational program.
4) Identify four types of software which could be used in CBI by describing activities that they would apply to in vocational programs.
5) Select software programs which you would use to supplement your instruction for developing:
    a) psychomotor skills
    b) affective behavior skills
    c) cognitive skills
    d) problem-solving skills
6) Develop a set of lesson plans designed to provide an outline for teaching five different applications of CBI in vocational programs.

## REFERENCES

CELL DEFENSE:
Hesware
150 North Hill Drive
Brisbane, CA 94005

CHIEF EXECUTIVE SERIES:
Lewis Lee Corporation
P.O. Box 51831
Palo Alto, CA 94303

KNOWARE AT HOME:
Knoware, Inc.
301 Vassar Street
Cambridge, MA 02139

LOTUS 123:
Lotus Development Corporation
161 First Street
Cambridge, MA 02142

MAESTRO:
Aquila Corporation
24 Park Street
Pepperell, MA 01463

MASTERTYPE:
Scarborough Systems, Inc.
25 North Broadway
Tarrytown, NY 10591

MULTIPLAN:
Microsoft
10700 Northup Way
Cambridge, MA 02142

PEACHCALC:
Peachtree Software Incorporated
3445 Peachtree Road N.E.
Atlanta, GA 30326

PEACHTEST 5000:
Peachtree Software Incorporated
3445 Peachtree Road N.E.
Atlanta, GA 30326

PERFECT WRITER:
Perfect Software
1001 Czmelia Street
Berkeley, CA 94710

THE SALES EDGE:
Human Edge Software
2445 Faber Place
Palo Alto, CA 94303

SCRIPSIT:
Tandy Corporation
Radio Shack
1550 One Tandy Center
Fort Worth, TX 76102

SPELLING BEE:
Munn Educational Software
P.O. Box 204
Groton Long Point, CT 06340

TESTMASTER:
Midwest Software
Box 214
Farmington, MI 48024

THWART!:
Random House Software
201 East 50th Street
New York, NY 10022

VISICALC:
Visicorp
2895 Zanker Road
San Jose, CA 95131

WORDPERFECT:
Satellite Software International
288 West Center Street
Orem, UT 84057

WORDSTAR:
Micropro International Corporation
1299 4th Street
San Rafael, CA 94901

*Microcomputer Software Programs for Vocational Education,* Rodenstein & Lanbert (1983) is available from the following source:

Vocational Studies Center
1025 West Johnson Street, Room 964
School of Education
University of Wisconsin
Madison, Wisconsin 53706

## Unit 4

# Select Appropriate Software For Specific Instructional Purposes

### UNIT OBJECTIVE

Upon completion of this unit, the learner will be able to select appropriate software for specific instructional purposes. This knowledge will be demonstrated through completion of the unit achievement indicators.

## SPECIFIC OBJECTIVES

Upon completion of this unit, the reader will be able to:

1) Explain the importance of selecting appropriate educational software.
2) Discuss the benefits for using appropriate microcomputer software in a teaching/learning situation.
3) Describe criteria for evaluating software for specific instructional purposes.
4) Evaluate instructional software for use in a specific teaching/learning situation.
5) Identify sources of software reviews for vocational educators.

# Select Appropriate Software For Specific Instructional Purposes

### BY: EMMA GEBO

Purchasing and using software developed by others can be a wise investment for vocational educators. For instructors, an important selection criterion is the appropriateness of software to specific instructional purpose.

The educational software market is changing faster than any other part of the software industry. At the time this unit is being written, the trend seems to be away from drill and practice, simple text displays, and games, and toward such uses as graphics, interaction, and simulation. Computers can be used to help test and assess student progress or to take corrective action if a student has not grasped an important concept. "Branching" software programs can be utilized to teach students at their individual "levels."

Each vocational educator should recognize that excellent educational software can turn a computer into a friendly, encouraging, and patient teacher (D'Ignazio, 1984). Quality educational software can be a significant aid to the teaching process and can provide instructors with more time to devote to individualized instruction. Vocational teachers should be critical and demanding of educational software. Before selecting a program, educators should read about it, talk to people who know it, and make sure it is the right computer program for their instructional objectives (D'Ignazio, 1984).

Selecting appropriate software for specific, instructional purposes requires an instructor's thought. However, the task of software selection is worth the required time and thought once teachers have identified specific instructional purposes. It would be senseless to have a microcomputer for a vocational program and not be able to use it for instructional purposes. The importance of educational software to computer-based instruction is paramount; without software, a computer is little more than a hunk of plastic and silicon that might as well be used as a doorstop (Taylor, 1984).

Vocational teachers must realize that educational software which is not oriented to users is virtually worthless. Teachers and students must be able to use software for their intended purposes without having to make expensive, time-consuming changes or adaptations. It is critical that software programs be analyzed for appropriate integration into the educational setting, and that vocational programs not be changed merely to meet the requirements of software.

Educators need to be assured that microcomputer software are meaningful inclusions in vocational curricula. It is essential for educators to define their instructional objectives and to identify those objectives which the microcomputer can help attain. The software then selected must be compatible with curriculum and in agreement with the philosophy of the vocational program. Educational software should feature clear and achievable goals, carefully designed reinforcement and feedback patterns, correct spelling and grammar, and content appropriate to the level and needs of the user. The characteristics and needs of the student population should be critical factors in the evaluation of software which is intended to meet specific instructional purposes.

Conkling (1983) describes the advent of Computer-Based Instruction in a county school system. The search for an innovative instructional system resulted in the implementation of Computer-Based Instruction. Following a period of use, students in the system were observed to be just as eager to sit down at the computer terminal in May as they were in September when the program was initiated. An attitude survey of students revealed that Computer-Based Instruction received the highest rating of any activity which involved academic instruction. Conkling stresses the importance of careful planning as a major reason for the success of this program. It is essential for vocational educators to identify the appropriate uses for microcomputer software in any instructional program. The implementation and integration of Computer-Based Instruction in an educational program can only be effective if educators learn how to implement hardware and software and then maximize their usage where they are most effective.

## The Needs of Intended Populations Must Be Considered

Little has been written about the effects of computers on the ways students learn and the manner in which they perceive their own abilities. Both the cognitive (intellectual, problem-solving) and affective (social, emotional) natures of microcomputer and user interaction seem to have a significant impact on the type and quality of learning that occurs with any one person (Ryba & Chapman, 1983). The perceptions of users concerning their personal control and effectiveness developed by use of the computer may result in psychological benefits to the student. These feelings include an improved self concept, reduction of emotional dependence, and self-management of behavior and learning.

Ryba and Chapman (1983) observe that mentally retarded students attain scores in microcomputer games that would be the envy of other students. Students have demonstrated persistence in understanding and mastering rules; have developed skills in planning, anticipating, sequencing, and executing combinations of movements; and have increased motor speed and coordination. The authors emphasize that much work is yet to be done in understanding how people learn and how microcomputers can assist learners. They observed an adult who had been labeled severely retarded and unlikely to benefit from formal training, but who was nevertheless playing a microcomputer game which required moving a cursor purposefully and making judgements about the order in which specific operations should be performed. Their contention is that this discrepancy raises the possibility that some students could be underfunctioning through lack of opportunity to demonstrate and improve their cognitive abilities with the aid of learning devices such as computers. These types of observations add fuel to the arguments against using standardized paper-and-pencil intellectual assessments (Ryba & Chapman, 1983).

The availability of microcomputers is resulting in increased exploration with computers in the instruction of handicapped learners. The individualized nature of microcomputer instruction is identified as one of the greatest potential benefits to handicapped persons. This benefit is especially true if the software developed and utilized are designed to branch to the level of instruction appropriate for the individual (Hannaford & Taber, 1982).

An exciting aspect of using microcomputers with vocational special needs students pertains to the capabilities of the machines to interact with learners. Microcomputers are able to present a stimulus, accept a response, evaluate the response, present appropriate feedback and reinforcement, and then move to the next appropriate instructional segment (Hannaford & Taber, 1982). This nonthreatening interaction can be especially beneficial to handicapped individuals who have faced failure or who have problems communicating. Autistic individuals and those with language impairments can particularly benefit from this interaction.

Obtaining software appropriate for use with vocational special needs students is not an easy task. Software may be written in the wrong computer language for a specific computer or the user may discover that more emphasis was placed on the technical aspects of the program than on the educational aspects. If microcomputer programs are to be effective with vocational special needs populations, both developers and purchasers of software must be sensitive to certain factors: (a) educational compatibility, (b) instructional design adequacy, and (c) technical adequacy (Hannaford & Taber, 1982). These three factors are mutually exclusive and should be individually considered by both developers and purchasers of software for instructional use with vocational special needs learners.

Disabled individuals who have, in the past, found themselves to be at a disadvantage are finding that they may not be at a disadvantage when it comes to working with microcomputers. Modifications to both hardware and software can be made to circumvent many physical disabilities. Touch screens, low typing tables, modems and speakerphones, and braille printouts are just a few of the modifications that have opened up the microcomputer world to disabled individuals. This can result in more independence and the possibility of new jobs for disabled individuals (Watt, 1984).

Educational software used in any teaching learning situation should be carefully evaluated and screened to be sure that it is not promoting bias in any form. The human factors involved in the program itself, as well as in the supplementary materials, are important. Vocational teachers should develop and use instructional materials which address a variety of racial and ethnic groups, as well as disabled persons and a range of ages. All instructional materials, including software, should be evaluated with criteria for bias (Rose, 1984).

## Evaluating Software for Suitability

Many articles have been written to help educators evaluate software programs. Some school districts, universities, and educational agencies have developed software evaluation tools for internal use. These instruments have developed to help educators evaluate or obtain an evaluation of software in an efficient manner. There is no single, industry-wide, evaluation method or tool that is used to evaluate software. However, several authors have developed guidelines for evaluating software. Savitsky (1984) summarizes ten criteria for judging the quality of software:

1. Has significant new content or skill-producing strategies.
2. Motivates and offers some challenge.
3. Guarantees the learner an emotionally healthy and appropriate learning environment.
4. Has educational objectives that are carefully chosen and clearly stated.
5. Has been subjected to rigorous field testing with appropriate learners.
6. Provides the learner with ease of control over the elements of the learning experience.
7. Provides a trail to permit the teacher/adult to monitor and review the learner's activities.
8. Provides clear documentation about all aspects of the program for the learner and the teacher.
9. Is accurate in the presentation of all facts and precise in the use of spelling, grammar, and usage.
10. Is free of personal abuse, sarcasm, derogatory or sexist/racist remarks.

With software for educational purposes being offered by so many different firms, the educator frequently wonders, "Is this program of high quality and will it be a real asset in the classroom?" Admittedly, some software on the market is not of high quality. The software may have been developed by people who know much about computers, but little about teaching. Or software might have been developed by someone with subject-matter and classroom experience, but little knowledge of the capabilities of the computer. Quality educational software is developed by those who have a combination of subject-matter and classroom experience as well as computer expertise.

An article in the January, 1983, issue of *Forecast for Home Economics* (p. 16) includes a list of 7 guidelines for quality educational software. The guidelines are based on a consensus of educators who use computers in the classroom, taken from *Electronic Learning* magazine:

1. **Be free of technical and educational errors.** The program should load properly and run smoothly. The information should be accurate, up-to-date, and educationally sound.
2. **Take advantage of the computer's unique capabilities.** A good computer program should be better than the traditional method of teaching the material. It should take advantage of the computer's capabilities and do some things that can't be done as well by a teacher or a text.
3. **Provide positive reinforcement and also help students to understand why their answers were wrong.** The students should feel good about themselves even if they answer incorrectly. While the program shouldn't reinforce, reward, or encourage wrong answers, it should help the students understand why their answer was wrong and then provide a concept, rule, or information that will help them to answer correctly.
4. **Include diagnostic and branching features.** Basically the program should make an attempt to determine the student's level of learning and then adapt to it. A pre-test will allow the program to branch to the student's level.
5. **Be creative, stimulating creativity among its users.** A good program should allow students to explore and ask questions and be open-minded. Students should not be locked into one right answer.
6. **Allow for easy teacher modification.** The program design should be such that the teacher can modify or change it to meet the specific needs of students.
7. **Provide clearly written support materials and activities.** The operating manual or other documentation should be written in language readily understood by the user, not computer jargon. Program prompts should be utilized so that the user does not have to keep referring to the manual to find out what to do next. Student worksheets and suggestions for supplementary activities are parts of the most valuable educational programs.

After an instructor has obtained what appears to be promising software, a decision must be made whether or not to buy the product. The criteria and guidelines presented in this unit must be considered as the teacher actually reviews or tries out a software program. As instructors review software, they must remember that the software selected must relate to the identified vocational curriculum objectives.

Vocational teachers can follow six steps that will help them in the software selection process (Kansky, Heck, & Johnson, 1981):

Step 1. Load the program into your computer system and run it briefly to become familiar with its "flow."

Step 2. Execute the program as a successful student would, avoiding intentional or careless errors. Extend the program when possible by interacting "creatively" as a good student would do in testing the cleverness of the programmer.

Step 3. Execute the program as an unsuccessful student would. Respond incorrectly to learn how the program handles student errors. If an erroneous response to a task results in the repetition of that task, make repeated incorrect responses. Be certain to repeat the same responses and also try giving different incorrect responses. Finally, make other kinds of errors such as typing mistakes, incorrect input types (e.g., "one" for "1"), content errors, and errors in following directions.

Step 4. Using criteria you've adopted, record your observations using some instrument such as a checklist.

Step 5. Compare your observations with the claims made by the vendor.

Step 6. Make a decision.

A software evaluation form is included in this unit (Neuman, 1982, pp. 45-48). A review of the form will verify that criteria identified for the evaluation of instructional software are consistent with guidelines presented in this unit. The emphasis is on the educational content presented, the manner in which the content is presented, interaction between the student and the computer, and the manner in which the teacher can use the program to complement the instructional program. Additional information on software evaluation is contained in unit B.5.

# SOFTWARE EVALUATION FORM

Reviewer's Name: ______________________ Date of Review: __________
Address: ______________________ Telephone: __________

Program Title: ______________________
Medium: ___ 5" disk ___ 8" disk ___ cartridge ___ tape
Package Title: ______________________
Copyright Date (if any) ______________________
Microcomputer (brand, model, memory) ______________________
Necessary Hardware __________ Necessary Software __________
Producer __________ Author(s) __________
Back-up Policy __________ Cost __________

## Part 1 - Program Overview and Description

1. Subject area and specific topic ______________________
2. Prerequisite skills necessary ______________________
3. Appropriate grade level(s) [circle] 1 2 3 4 5 6 7 8 9 10 11 12 College
4. Type of program (check one or more)

___ Simulation
___ Educational Game
___ Drill and Practice
___ Tutorial
___ Problem Solving
___ Authoring System
___ Testing
___ Classroom Management
___ Remediation
___ Enrichment
___ Other (specify) ______________________

5. Appropriate group instructional use of the computer? ______________________
6. Is this program an appropriate instructional use of the computer? __________
7. Briefly list the program's objectives. Are they clearly stated in the program or in the documentation? Are they educationally valuable? Are they achieved? ______________________
8. Briefly describe the program. Mention any special strengths or weaknesses.

## Part 2 - Evaluation Checklist

Please check YES, NO, or NOT APPLICABLE for each question below. To add information, or to clarify an answer, use the COMMENTS space at the end of each section.

| EDUCATIONAL CONTENT | YES | NO | N/A |
|---|---|---|---|
| 1. Is the program content accurate? | | | |
| 2. Is the program content appropriate for intended users? | | | |
| 3. Is the difficulty level consistent for material, interest, vocabulary? | | | |
| 4. Is the program content free of racial, sexual, or political bias? | | | |
| COMMENTS: | | | |

| PRESENTATION | YES | NO | N/A |
|---|---|---|---|
| 1. Is the program free of technical problems?<br>2. Are the instructions clear?<br>3. Is the curriculum material logically presented and well organized?<br>4. Do graphics, sound, and color, if used, enhance the instruction?<br>5. Is the frame display clear and easy to read? | | | |
| COMMENTS: | | | |

| INTERACTION | YES | NO | N/A |
|---|---|---|---|
| 1. Is the feedback effective and appropriate?<br>2. Do cures and prompts help students to answer questions correctly?<br>3. Can students access the program "menu" to help or change activities?<br>4. Can students control the pace and sequence of the program?<br>5. Are there safeguards against the students bombing the program? | | | |
| COMMENTS: | | | |

| TEACHER USE | YES | NO | N/A |
|---|---|---|---|
| 1. Is record-keeping possible (within the program or through documentation worksheets)?<br>2. Does the teacher have to monitor student use?<br>3. Can the teacher modify the program?<br>4. Is the documentation clear and comprehensive? | | | |
| COMMENTS: | | | |

## Part 3 - Overall Evaluation

Check one:

___ Excellent program. Recommend without hesitation.
___ Pretty good program. Consider purchase.
___ Fair. But might want to wait for something better.
___ Not useful. Do not recommend purchase.

**Where Can Vocational Teachers Learn More About Educational Software?**

There are many sources that vocational educators can utilize to help identify software, review software, and suggest appropriate uses for software. Rapid changes in software design and the volume of software available for instructional purposes make it almost impossible to keep up with the task of evaluating software. Three major sources are recommended for vocational instructors for obtaining software evaluation data (Holznagel, 1983):

1. The MicroSIFT project at the Northwest Regional Educational Laboratory (NWREL). Through this project, evaluation reports are sent to state and regional centers and school districts, who then make copies available to their constituents. The data obtained are also added to the software section of the REIC (Resources in Computer Education) database. This database is designed and maintained by the NWREL.
2. The EPIE Institute is the second major source. EPIE also uses a standard form which utilizes predefined criteria and open-ended comments. The evaluation reports, published as *Micro-Courseware PRO/FILES,* may be available in your state through the state department of education.
3. Professional journals and magazines in computer education comprise the third major source for software evaluative data. The *AEDS Monitor, ICCE Computing Teacher,* and *Electronic Learning* are three sources. The *Courseware Report Card* is a periodical dedicated to courseware evaluation reports.

It is important to consult a variety of sources when considering the purchase of a software package. Evaluations are generally based on the opinions of several evaluators; since opinions may differ, it is wise to consult as many sources as possible for information (Holznagel, 1983).

Although producers of software have been reluctant to allow potential purchasers to preview or try out software prior to purchase, this policy may be changing. Distributors are developing methods for prospective buyers to evaluate software prior to its purchase.

Software developers are producing incomplete reproductions of retail products for potential buyers to review. These incomplete products present samples of the total program for the customer to review prior to purchase. This practice should reduce the risk of a regrettable purchase. Some products of sample software encourage copying of the samples, in the belief that this policy will spread advertising of the product. Some samples are developed so that a prospective buyer may develop a file with data while testing the sample, and the file can then be transferred to the program disk. Purchasers should be cautioned that a demo disk may not be able to provide a realistic look at the scope or actual operation of a program (Watt, 1984).

## SAMPLE LISTING OF SOFTWARE VENDORS

The following list represents a sampling of a much greater number of existing software producers. Vocational educators can use this list as a beginning point for searching for instructional software. Several of these vendors will provide teachers with free catalogs of their software products (Zahniser, Long, & Nasman, 1983, pp. 29-40).

Abbott Educational Software
334 Westwood Ave.
East Longmeadow, MA 01028
(413) 525-3462

Adventure International
P.O. Box 3435
Longwood, FL 32750
(305) 862-6917

American Analysis Corporation
655 Redwood Highway
Mill Valley, CA 94941

Applied Educational Systems
RFD 2, Box 213
Dunbarton, NH 03301
(603) 774-6151

Atari
call (800) 538-8547 for name
of nearest dealer

Acorn Software Products
634 North Carolina Ave. S.E.
Washington, DC 20003
(202) 544-4259

Alternate Source
1806 Ada Street
Lansing, MI 48910
(517) 487-3358

Apple Computer Company
10260 Bandley Drive
Cupertino, CA 95014
(408) 996-1010

Applications
21650 W. Eleven Mile Road
Suite 103
Southfield, MI 48076

Robert R. Baker, Jr.
5845 Topp Court
Carmichael, CA 95608

Basics & Beyond
P.O. Box 10
Amawalk, NY 10501
(914) 962-2355

Bell & Howell Micro Systems
Audio Visual Products Division
7100 North McCormick Road
Chicago, IL 60645
(312) 262-1600

Borg-Warner Educational Systems
600 West University Drive
Arlington Heights, IL 60004
(800) 323-7577

Business Microproducts
1838 Catalina Court
Livermore, CA 94550

Cavri Systems
26 Trumbull Street
New Haven, CT 06511
(203) 562-9873

Comm Data Systems
P.O. Box 325
Milford, MI 48042

COMPress
P.O. Box 102
Wentworth, NH 03282
(603) 764-5831

Computer Information Exchange
P.O. Box 159
San Luis Rey, CA 92068
(714) 757-4849

Conduit
P.O. Box 388
Iowa City, IA 52244
(319) 353-5789

Courseware Magazine
4919 North Millbrook 222
Fresno, CA 93726
(209) 225-0953

Creative Computing Software
Department 301
P.O. Box 789M
(201) 540-0445

Dallas Public Schools
Marketing Department
912 South Ervay
(214) 742-7991

Demi-Software
6 Lee Road
Medfield, MA 02052

BCD Associates, Inc.
1216 North Blackwelder
Oklahoma City, OK 73106
(405) 524-7403

Betamax, Inc.
101 Nickerson Street
Suite 550
Seattle, WA 98109
(206) 282-6249

The Bottomshelf, Inc.
P.O. Box 49104
Atlanta, GA 30359

Carta Associates, Inc.
640 Lancaster Ave.
Frazer, PA 19355
(215) 647-9600

Comaldor
25 Sunrise Ave. 1108
Toronto, Ontario, Canada M4A 2S2
phone: 751-7481

Commodore Business Machines
(contact local Commodore dealer)

Computer Business Systems of
Myrtle Point
1707 Viewpoint
Myrtle Point, OR 97458
(503) 572-3841

Computrex Computer Services
P.O. Box 536
Inman, SC 29349

Cook's Computer Company
1905 Bailey Drive
Marshaltown, IA 50158

Cow Bay Computing
Box 515
Manhasset, NY 11030

Curriculum Applications
P.O. Box 264
Arlington, MA 02174

Delmarva Computer Club
P.O. Box 36
Wallops Island, VA 23337

Desert Sound
16268 Main Street
Hesperia, CA 92345
(714) 244-2555

Duxbury Systems, Inc.
77 Great Road
Action, MA 01720

Educational Activities, Inc.
P.O. Box 392
Freeport, NY 11520
(516) 223-4666

Educational Microsystems, Inc.
P.O. Box 471
Chester, NJ 07930
(201) 879-5982

Educational Services Management
P.O. Box 12599
Research Triangle Park, NC 27709
(919) 781-1500

Educational Software and Design
P.O. Box 2801
Flagstaff, AZ 86003

Eduteck Corp.
P.O. Box 11354
Palo Alto, CA 94366
(415) 325-9965

Edu-Ware Services, Inc.
22222 Sherman Way
Suite 102
Canoga Park, CA 91303
(213) 346-6783

Fireside Computing, Inc.
MicroGonome Division
5843 Montgomery Road
Elkridge, MD 21227
(301) 796-4165

Goforth Microcomputing
329 22nd Street East
Prince Albert, Saskatchewan,
Canada S6V IN3
(306) 763-8323

Hartley Courseware, Inc.
P.O. Box 431
Diamondale, MI 48821
(616) 942-8987

High Technology Software
Products, Inc.
P.O. Box 14665
Oklahoma City, OK 73113
(405) 840-9900

Ideatech
P.O. Box 62451
Sunnyvale, CA 94088

George Earl
1302 South General McMullen
San Antonio, TX 78237

Educational Courseware
Ten Bay Street
Design 66
Westport, CT 06880

Educational Programs
P.O. Box 2345
West Lafayette, IN 47906
(317) 463-4778

Educational Software Corp.
414 Rosemere
Maquoketa, IA 52060

EduTech
50 Putnam Street
West Newton, MA 02165
(617) 965-4813

Edu-Ware
P.O. Box 336
Maynard, MA 01754

Ellis Computing
600 41st Ave.
San Francisco, CA 94121
(415) 751-1522

Gentech Corp.
4101 N. St. Joseph Ave.
Evansville, IN 47712
(812) 423-4200

J. L. Hammett
P.O. Box 545
Hammett Place
Braintree, MA 02184

Hayden Book Company
50 Essex Street
Rochelle Park, NJ 07662
(800) 631-0856

Houghton Mifflin Company
One Beacon Street
Boston, MA 02103
(617) 725-5000

Indian Head Software
1002 Indian Head Drive
Snow Hill, NC 28580
(919) 747-2839

Instant Software
Peterborough, NY 03458
(800) 258-5473

International Micro Systems
8425 Quivara Road
Lenexa, KS 66215
(913) 888-8330

Jem Research
Discovery Park
P.O. Box 1700
University of Victoria
Victoria, BC Canada V8W 2Y2

Krell Software
21 Millbrook Drive
Stony Brook, NY 11790
(518) 751-5139

Little Bee Educational Programs
P.O. Box 262
Massillon, OH 44648
(216) 832-4097

Mastertype
P.O. Box 5223
Stanford, CA 94305

Math Software
1233 Blackthorn Plaza
Deerfield, IL 60015

Med Systems Software
P.O. Box 2674-D
Chapel Hill, NC 27514
(919) 933-1990

Mentor Software, Inc.
P.O. Box 791
Anoka, MN 55303

Microcomputer Education
Applications Network
256 N. Washington Street
Falls Church, VA 22046
(702) 536-2310

MICRO-ED, Inc.
P.O. Box 24156
Minneapolis, MN 55424
(612) 926-2292

Micro Learningware
P.O. Box 2134
North Mankato, MN 56001
(507) 625-2205

Micro Power & Light
13773 North Central Expressway
Dallas, TX 75243
(214) 234-8233

Instructional Development Systems
2927 Virginia Beach Blvd.
Virginia Beach, VA 23452
(804) 340-1977

Interpretive Education
2306 Winters Drive
Kalamazoo, MI 49002
(616) 345-8681

Jensen Software
1440 Rockway
Lakewood, OH 44107

Library Software
P.O. Box 23897
Pleasant Hill, CA 94523

Charles Mann & Associates
Microcomputer Division
55722 Santa Fe Trail
Yucca Valley, CA 92284
(714) 365-9718

Math City
4040 Palos Verdes Drive North
Rolling Hills Estates, CA 90274
(213) 541-3377

McGraw-Hill
1221 Avenue of the Americas
New York, NY 10029
(212) 997-1221

Mega-byte Systems
66 Church Street
Ellenville, NY 12428
(914) 647-4235

Merlan Scientific
P.O. Box 25
Depew, NY 14043
(416) 877-0171

Microcomputer Software Systems
4716 Lakewood Drive
Metairie, LA 70002

micro lab
811 Stonegate Drive
Highland Park, IL 60035
(312) 433-7877

Microphys
2048 Ford Street
Brooklyn, NY 11229
(212) 646-1040

Micropute Software
P.O. Box 1943
Rock Mount, NC 27801

Microsoft Consumer Products
10800 North East Eighth
Bellevue, WA 98004
(206) 454-1315

Milton-Bradley Company
Shaker Road
E. Longmeadow, MA 01028
(413) 525-6411

Mosaic Electronics
P.O. Box 748
Oregon City, OR 97045

NCCD (National Coordinating
Center for Curriculum Development)
State University of New York
Stony Brook, NY 11794
(516) 246-8418

OMNICO Computer Associates
3300 Buckeye Road
Atlanta, GA 30341

Program Design, Inc.
11 Idar Court
Greenwich, CT 06830
(203) 661-8799

Project COMCAL
Commack Public Schools
Hauppage Road
Commack, NY 11725

Quality Educational Designs
P.O. Box 12486
Portland, OR 97212
(503) 282-4906

Rainbow Micro Software
1650 Piikea Street
Honolulu, HI 96818
(TRS-80 Programs)

School Microware
P.O. Box 246
Dresden, ME 04342
(207) 737-4466

Shafer Software
465 South Mathilda Ave.
Suite 202
Sunnyvale, CA 94086
(408) 730-0179

The Software Exchange
6 South Street, Box 68
Milford, NH 03055
(603) 673-5144

Millikein Publishing Company
1100 Research Boulevard
St. Louis, MO 63132
(314) 991-4220

Minnesota Educational Computing
Consortium Publications
2520 Broadway Drive
St. Paul, MN 55113
(800) 631-8112

MUSE Software
330 North Charles Street
Baltimore, MD 21201
(301) 659-7212

National Software Marketing
4701 McKinley Street
Hollywood, FL 33021
(305) 625-6062

Petsoft
Radclyffe House
66-68 Hagley Road
Birmingham, England B16 8PF

Programs for Learning, Inc.
P.O. Box 954
New Milford, CT 06776

Project LOCAL Software
c/o Dresden Associates
P.O. Box 246
Dresden, ME 04342
(207) 737-4466

(Radio Shack - sold through local
Radio Shack retail outlets)

Scharf Software Services
P.O. Box 18445
Irvine, CA 92713
(714) 557-9206

Science Research Associates
155 North Wacker Drive
Chicago, IL 60606
(800) 621-0664

Sheridan College
c/o F. Winter
1439 Trafalgar Road
Oakville, Ontario
Canada L6H 2L1
(416) 845-9430

Software Industries
902 Pinecrest
Richardson, TX 75080

Special Delivery Software
10260 Bandley Drive
Cupertino, CA 95041
(408) 996-1010

The Teaching Assistant
22 Seward Drive
Huntington Station, NY 11746

T.H.E.S.I.S.
P.O. Box 147
Garden City, MI 48135
(313) 595-4722

Teacher's Pet
c/o Glenn Fisher
1517 Holly Street
Berkeley, CA 94703

Tycom Associates
68 Velma Avenue
Pittsfield, MA 01201
(413) 442-9771

Unicom Division of United Camera
297 Elmwood Avenue
Providence, RI 02907

Whitney Educational Services
2071 Tenth Ave.
San Francisco, CA 94116

Stoneware Microcomputer Products
1930 Fourth Street
San Rafael, CA 94901
(415) 454-6500

Teaching Tools
P.O. Box 12679
Research Triangle Park, NC 27709
(919) 851-2374

3R Software
P.O. Box 3115
Jamaica, NY 11341

Time Share Corp.
Hanover, NH 03775
(603) 488-3838

TYC Software
40 Stuyvesant Manor
Geneseo, NY 14454
(716) 243-3005

Westinghouse Learning Corp.
5005 West 110th Street
Oak Lawn, IL 60453

Rodenstein & Lanbert (1983) authored *Microcomputer Software Programs For Vocational Education*, a document highly recommended for vocational instructors who are just beginning the search for software for vocational programs. It is one of the few resources solely devoted to software specific to vocational education. Software is categorized in the document by the name of the program, its distributor and the vocational discipline that it represents. As examples, the following excerpts from the document list software for distributive education (Rodenstein & Lanbert, 1983, pp. 35-36) and trade and industrial education (pp. 32-34).

## DISTRIBUTIVE EDUCATION—GENERAL

| **Name of Program** | **Distributor** |
|---|---|
| PAYFILE<br>(for small businesses) | Dynacomp, Inc., 1427 Monroe Avenue, Rochester, NY 14618 (800) 828-6772 |
| SALE I<br>(sales analysis) | Micro Architect, Inc., 96 Dotha Street, Arlington, MA 02174 |
| CALC I<br>(business package) | |
| COSTING | Computer House Division, 1407 Clinton Road, Jackson, MI 44202 (517) 782-2132 |
| RETAIL MANAGEMENT SYSTEM | Charles Mann and Associates, 7594 San Remo Trail, Yucca Valley, CA 92284 (714) 365-9718 |
| PORTFOLIO | Southeastern Educational Software 3300 Buckeye Rd., Atlanta, GA 30341 (404) 457-8336 |

| | |
|---|---|
| LEMONADE STAND | Creative Computing<br>Software, 39 E. Hanover<br>Ave., Morris Plains, NJ 07950 |
| CARTELS & CUTTHROATS | Strategic Simulations<br>465 Fairchild Dr., Suite<br>108, Mountain View, CA<br>94043 |
| CONGLOMERATES COLLIDE | Rockroy, 7721 E. Gray<br>Rd., Suite 103<br>Scottsdale, AZ 85260 |
| MANAGEMENT DECISIONS | Hayden Book Company<br>50 Essex Street<br>Rochelle Park, NJ 07662 |
| MARKET | Creative Computing<br>Software, 39 E. Hanover<br>Ave., Morris Plains, NJ 07950 |
| MICRO MILLIONAIRE | Softside, 6 South St.<br>Milford, NH 03055 |
| PRODUCTION COSTING | Curriculum Publications<br>Clearinghouse, Western<br>Illinois Univ., 46<br>Horrabin Hall, Macomb,<br>IL 61455 |
| INVENTORY ANALYSIS - REORDER POINT<br>- Projected use<br>- Turnover rate | Curriculum Publications<br>Clearinghouse, Western<br>Illinois Univ., 46<br>Horrabin Hall, Macomb,<br>IL 61455 |
| JOB PRICE/BIDDING | Curriculum Publications<br>Clearinghouse, Western<br>Illinois Univ., 46<br>Horrabin Hall, Macomb<br>IL 61455 |
| FORECASTING - LEAST SQUARES<br>- Regression<br>- Moving Average<br>- Exponential Smoothing | Curriculum Publications<br>Clearinghouse, Western<br>Illinois Univ., 46<br>Horrabin Hall, Macomb,<br>IL 61455 |

## TRADE AND INDUSTRIAL

| **Name of Program** | **Distributor** |
|---|---|
| GENERAL APPLE DISKETTE ON INDUSTRIAL ARTS | Minnesota Curriculum Services<br>Center, 3554 White Bear Ave.<br>White Bear Lake, MN 55110<br>(612) 770-3943 |
| THERMODYNAMICS | MicroLearningWare, P.O. Box 2134<br>North Mankato, MN 56001<br>(507) 625-2205 |
| ELECTRICITY | MicroLearningWare, P.O. Box 2134<br>North Mankato, MN 56001<br>(507) 625-2205 |

| | |
|---|---|
| MACHINE PART QUOTING | Computer House Division, 1407<br>Clinton Road, Jackson, MI 44202<br>(517) 782-2132 |
| PRINT SHOP JOB ESTIMATING/ SCHEDULING | California Micro Products, 795<br>West Imperial Highway, Brea, CA<br>92621 (714) 990-4014 |
| CONSTRUCTION COST/PROFIT ANALYSIS | Realth Software, 2045 Manhattan<br>Ave., Hermasa Beach, CA 95014<br>(408) 996-1010 |
| DESIGNER TOOL KIT (drafting) | Realth Software, 2045 Manhattan<br>Ave., Hermasa Beach, CA 95014<br>(408) 996-1010 |
| CACTUS PLOT | Southwest Educational Psychology<br>Services, Inc., P.O. Box 1870<br>Phoenix, AZ 85001<br>(602) 253-6528 |
| ELECTRONIC ENGINEER'S ASSISTANT | Opportunities for Learning, Inc.<br>8950 Lurline Ave., Dept. 390<br>Chatsworth, CA 91311<br>(213) 341-2535 |
| ELECTONICS I | Opportunities for Learning, Inc.<br>8950 Lurline Ave., Dept. 390<br>Chatsworth, CA 91311<br>(213) 341-2535 |
| PRACTICAL DIGITAL THEORY & TROUBLESHOOTING TIPS | Opportunities for Learning, Inc.<br>8950 Lurline Ave., Dept. 390<br>Chatsworth, CA 91311<br>(213) 341-2535 |
| UNI-SOLVE | Opportunities for Learning, Inc.<br>8950 Lurline Ave., Dept. 390<br>Chatsworth, CA 91311<br>(213) 341-2535 |
| HAM PACKAGE | Opportunities for Learning, Inc.<br>8950 Lurline Ave., Dept. 390<br>Chatsworth, CA 91311<br>(213) 341-2535 |
| 50 HI-RES ELECTRONIC DESIGN | Avant-Garde Creations, P.O.<br>Box 30160, Eugene, OR 97403<br>(503) 345-3043 |
| TEACH YOURSELF MICROPROCESSORS | Integrated Computer Systems<br>3304 Pico Blvd., P.O. Box 5339<br>Santa Monica, CA 90405 |
| HI-TECH EDUCATIONAL EQUIPMENT<br>-microprocessor systems control systems, applications modules, CNC machines, interface boards, instrumentation, and control | Feedback, 620 Springfield<br>Avenue, Berkeley Heights, NJ 07922<br>07922 (201) 464-5181 |
| ROBOTICS<br>-Armover, Robert Armborg, Genesis Armatrol, Armadilo, Armsort, Armdraulic | Feedback, 620 Springfield<br>Avenue, Berkeley Heights, NJ<br>07922 (201) 464-5181 |

| | |
|---|---|
| SHORT CIRCUITS | Keenan Educational Games, Inc.,<br>95 Ponderosa Dr., Hanover, MA<br>02339 (617) 878-7341 |
| SUPER SHORT CIRCUITS | Keenan Educational Games, Inc.,<br>95 Ponderosa Dr., Hanover, MA<br>02339 (617) 878-7341 |
| EDUCATION TRAINING PACKAGES | E & L Instruments, Inc., 61 First<br>Street, Derby, CT 06418<br>(203) 735-8774 |
| HICKOK TEACHING SYSTEMS<br>-Hickok Micro Trainer<br>software, courseware, &<br>are for digital &<br>ter technology | Allison Associates, Box 313,<br>Troy, MI 48099<br>(313) 689-2990 |
| TEACHMOVER<br>(developmental robotic arm) | Allison Associates, Box 313,<br>Troy, MI 48099<br>(313) 689-2990 |
| MINIMOVER—5 GRIPPER | Allison Associates, Box 313,<br>Troy, MI 48099<br>(313) 689-2990 |
| TEN POSITION ROBOTICS LABORATORY | Allison Associates, Box 313,<br>Troy, MI 48099<br>(313) 689-2990 |
| FIVE POSITION ROBOTICS LABORATORY | Allison Associates, Box 313<br>Troy, MI 48099<br>(313) 689-2990 |
| MICROBOT ALPHA | Allison Associates, Box 313<br>Troy, MI 48099<br>(313) 689-2990 |
| LENNOX ROBOTICS TRAINING PROGRAMS & TRAINING AIDS | Lennox Education Products<br>P.O. Box 400450, Dallas, TX<br>75240 (214) 783-5463 |
| SINGLE ARM SLIDE ROBOT | Lennox Education Products<br>P.O. Box 400450, Dallas, TX<br>75240 (214) 783-5463 |
| DOUBLE ARM SLIDE ROBOT | Lennox Education Products<br>P.O. Box 400450, Dallas, TX<br>75240 (214) 783-5463 |
| BOX ARM SLIDE ROBOT | Lennox Education Products<br>P.O. Box 400450, Dallas, TX<br>75240 (214) 783-5463 |
| STEPPER MOTORS ROBOT | Lennox Education Products<br>P.O. Box 400450, Dallas, TX<br>75240 (214) 783-5463 |
| ELECTRIC SERVO CONTROLLED ROBOT | Lennox Education Products<br>P.O. Box 400450, Dallas, TX<br>75240 (214) 783-5463 |
| HYDRAULIC SERVO CONTROLLED ROBOT | Lennox Education Products<br>P.O. Box 400450, Dallas, TX<br>75240 (214) 783-5463 |

| | |
|---|---|
| BASIC ELECTRIC TRAINER | Lennox Education Products<br>P.O. Box 400450, Dallas, TX<br>75240 (214) 783-5463 |
| RELAY TRAINER | Lennox Education Products<br>P.O. Box 400450, Dallas, TX<br>75240 (214) 783-5463 |
| MICROCOMPUTER APPLICATIONS IN VOCATIONAL EDUCATION: TRADES AND INDUSTRY | Western Illinois Univ.<br>Macomb, IL 61455 |
| CADAPPLE SOFTWARE | T & W Systems, Inc.<br>18437 Mt. Langley<br>Suite B, Fountain<br>Valley, CA 92708<br>(714) 963-3913 |
| CAT-1 COMPUTER AIDED DESIGN SYSTEM | Catoonix Corporation<br>151 Sixth St., N.W. Suite<br>039, Atlanta, GA 30313 |

## INDUSTRIAL EDUCATION

| **Name of Program** | **Distributor** |
|---|---|
| GENERAL APPLE DISKETTE ON INDUSTRIAL ARTS | Minnesota Curriculum Services Center<br>3554 White Bear Avenue, White Bear<br>Lake, MN 55110 (612) 770-3943 |
| THERMODYNAMICS<br>NORTH | MicroLearningWare, P.O. Box 2134, North<br>Mankato, MN 56001 (507) 625-2205 |
| MACHINE PART QUOTING | Computer House Division, 1407 Clinton<br>Road, Jackson, MI 44202<br>(517) 782-2132 |
| PRINT SHOP JOB ESTIMATING/ SCHEDULING | California Micro Products, 795 West<br>Imperial Highway, Brea, CA 92621<br>(714) 990-4014 |
| CONSTRUCTION COST/PROFIT ANALYSIS | Realth Software, 2045 Manhattan Ave.<br>Hermasa Beach, CA 90254 |
| CIRCUIT ANALYSIS | Apple Computer, 2045 Mariani Ave.<br>Cupertino, CA 95014<br>(408) 996-1010 |
| ELECTRICITY | MicroLearningWare, P.O. Box 2134,<br>North Mankato, MN 56001<br>(507) 625-2205 |
| CACTUS PLOT | Southwest Educational Psychology<br>Services, Inc., P.O. Box 1870,<br>Phoenix, AZ 85001 (602) 253-6528 |
| ELECTRONIC ENGINEERING ASSISTANT | Opportunities for Learning,<br>Chatsworth, CA 91311 (213) 341-2535 |
| ELECTRONICS I | Opportunities for Learning,<br>Chatsworth, CA 91311 (213) 341-2535 |
| DESIGNER TOOL KIT (drafting) | Apple Computer, 20525 Mariani Avenue<br>Cupertino, CA 95014<br>(408) 996-1010 |

# AGRICULTURE
# HERD PERFORMANCE PROGRAMS

| **Name of Program** | **Distributor** |
|---|---|
| CATTLE FEEDING ECONOMICS | AG-Com,<br>Muscatine, IA 52761 |
| COW-CALF PROFITABILITY | AG-Com,<br>Muscatine, IA 52761 |
| DAY-ADJUSTED WEANING WEIGHTS | AG-Com,<br>Muscatine, IA 52761 |
| FARROW TO FINISH SWINE PRODUCT | AG-Com,<br>Muscatine, IA 52761 |
| FINISHING FEEDER PIGS | AG-Com,<br>Muscatine, IA 52761 |
| NET ENERGY FOR FEEDLOT CATTLE | AG-Com,<br>Muscatine, IA 52761 |
| PROTEIN BALANCE FOR FEEDLOT | AG-Com,<br>Muscatine, IA 52761 |
| SHEEP PRODUCTION—ECONOMICS | AG-Com,<br>Muscatine, IA 52761 |
| SWINE RATION ANALYSIS | AG-Com,<br>Muscatine, IA 52761 |
| SWINE RATION FORMULATION | AG-Com,<br>Muscatine, IA 52761 |
| AGRICULTURE—VOLUME I DISKETTE | MECC, 2520 Broadway Drive, St. Paul, MN 55113 (612) 376-1118 |
| **Software in Cost Tracking** | |
| FERTILIZER, HERBICIDES, MARKETING, PLANTS BREAKEVENS, ANALYSIS, LEASES, WEATHER, DAIRYING, HARVESTING, IRRIGATION, INVENTORY, STORAGE, FUNGICIDES, INSECTS | AGPROS MicroSystems, Box 64539, Lubbock, TX 79464 (808) 745-3011 |
| FARM PROGRAMS | MicroLearningWare, P.O. Box 2134,<br>North Mankato, MN 56001<br>(507) 625-2205 |
| LIVESTOCK MANAGEMENT | MicroLearningWare, P.O. Box 2134,<br>North Mankato, MN 56001<br>(507) 625-2205 |
| BEST CROP, FINISHING HOG PLANNER, FEEDER SPECIFICS, PIG PRODUCTION PLANNER, BEEF FEEDER PLANNER, DAIRY COW PRODUCTION PLANNER, CASH FLOW SUMMARY, INVENTORY SUMMARY | F.A.R.M. Aids (for Apple II),<br>Specialized Data Systems, Box 8278<br>Madison, WI 53708 (608) 241-5050 |
| CATTLE BREEDING RECORDKEEPING SYSTEM FOR MODEL III | F.A.R.M. Aids (for Apple II),<br>Specialized Data Systems, Box 8278,<br>Madison, WI 53708 (608) 241-5050 |

*Microcomputer Software Programs for Vocational Education*, Rodenstein & Lanbert (1983) is available from the following sources:

Vocational Studies Center
1025 West Johnson Street, Rm 964
School of Education
University of Wisconsin
Madison, Wisconsin 53706

## SUMMARY

Although it may be tempting to obtain and use specific software because it has been developed by a particular company, or because it was obtained from another educator who liked it, DON'T! Vocational educators must evaluate microcomputer software for use in the specific instructional program based on the suitability and the appropriateness of the software for teaching specific instructional objectives. The software selected, purchased, and used in any vocational program must be suitable to the needs, interests, levels, and abilities of the students for whom it is intended.

Cognitive, affective, and psychomotor skills can be gained using appropriate software in an instructional setting. Research has shown this to be true with many different populations of students. The key appears to be to use programs that can "branch" to the educational levels of the students using the programs.

Evaluating software, although a time consuming process for the vocational educator, is an essential process. Criteria to consider, steps for evaluating a piece of software, and a specific evaluation tool are presented for the vocational educator in this unit of instruction. Utilization of these tools will help assure vocational educators that their decisions regarding software are intelligent and thorough.

Reviews of software by other individuals and sources can be a valuable aid for the vocational educator. These reviews should be regarded thoughtfully, keeping in mind that the reviews are the opinions of other individuals.

## ACHIEVEMENT INDICATORS

1) Explain why software selected by vocational teachers should be appropriate to the instructional setting.
2) Describe an example of a cognitive, affective, and psychomotor learning outcome that may result from the student using an appropriate software program in a teaching/learning situation.
3) List at least 8 criteria to consider when evaluating a piece of microcomputer software for inclusion in an instructional program.
4) State at least three sources of software reviews available to the educator.
5) Obtain a piece of software and evaluate it for inclusion in teaching a specific instructional objective. Use the evaluation form provided in this unit to review the software.

## REFERENCES

Computer Savvy: How to evaluate software. (1983). *Forecast for Home Economics, 28* (5), 16.

Conkling, R. (April, 1983). The nuts and bolts of selecting a computer assisted instructional program. *T.H.E. Journal*, 101-104.

D'Ignazio, F. (1983/1984). Education. *PC World Special Edition*, pp. 268-269.

Hannaford, A., & Taber, F. (1982). Microcomputer software for the handicapped: Development and evaluation. *Exceptional Children, 49* (2), 137-142.

Holznagel, D. (1983). Evaluating software. *AEDS Journal, 17* (1 & 2), 33-40.

Kansky, B., Heck, W., & Johnson, J. (1981). Getting hard-nosed about software: Guidelines for evaluating computerized instructional materials. *Mathematics Teacher, 74*, 600-604.

Neuman, A. (1982, October). The software lineup: What reviewers look for when evaluating software. *Electronic Learning*, pp. 46-48.

Rodenstein, J., & Lanbert, R. (1983). *Microcomputer Software Programs for Vocational Education*. Madison: University of Wisconsin.

Rose, R. (1984). Identifying equitable software. *The Computing Teacher, 11* (8), 51.

Ryba, K., & Chapman, J. (1983). Toward improving learning strategies and personal adjustment with computers. *The Computing Teacher, 11* (1), 48-53.

Savitsky, D. (1984). A guide to producing educational software. *The Computing Teacher, 11* (7), 26-27.

Taylor, A. (April, 1984). The wizard inside the machines. *Time*, pp. 56-63.

Watt, P. (March, 1984). Computers give independence to the disabled. *InfoWorld*, pp. 30-31.

Watt, P. (May, 1984). Try before you buy. *InfoWorld*, pp. 42-43.

Zahniser, G., Long, J., & Nasman, L. (1983). *An administrator's guide to microcomputer resources* (Research & Development Series No. 239B). Columbus, OH: The National Center for Research in Vocational Education.

## Unit 5

# Modifying Software

### UNIT OBJECTIVE

Upon completion of this unit, the learner will be able to modify software. This knowledge will be demonstrated through completion of the achievement indicators at the conclusion of this unit.

### SPECIFIC OBJECTIVES

Upon completion of this unit, the reader will be able to:

1) Explain four reasons why instructors may need to modify software.
2) Describe five fundamental rules of software modification.
3) Correctly use nine statements of the basic programming language.
4) Modify computer program listings of up to one page which may contain minor errors or be in need of improvement.

# Modifying Software

### BY: DR. ROBERT WATTS

```
]RUN LADDER ANGLE SAFETY
******** LADDER ANGLE SAFETY ********
LADDED BASE MUST NOT BE OUT FROM        THE HOUSE IN FEET:
                   NOT LESS THAN—NOT MORE THAN—LADDER LENGTH
                        8.71          9.32          36
                        8.23          8.8           34
                        7.74          8.28          32
                        7.26          7.76          30
                        6.77          7.25          28
                        6.29          6.73          26
                        5.81          6.21          24
                        5.32          5.69          22
                        4.84          5.18          20
```

Assume that the above is a program with minor errors. Although the program contains useful information for students, it needs correction. If the instructor corrects the spelling and improves the format, then this simple program will illustrate an important point and be useful in class.

The ladder safety example is simple to understand and to change. Vocational instructors may very well assume that the results are worth the time spent in making corrections. However, instructors will soon see that not all programs are so easily modified, so why bother?

Software modification is an interesting way to improve programming skills and to appreciate how others solve programming problems. Furthermore, a large quantity of good software exists in the public domain. Instructors may use, modify, or copy these programs to meet their instructional needs. Instructors will usually modify public domain software rather than use the $150.00 copyrighted commercial software.

There are four reasons why vocational teachers may wish to modify software:

1. The vocational teacher has a program which runs, but which includes an error.
2. The program runs satisfactorily and is correct, but the instructor would like to improve it or customize it for a specific teaching situation.
3. The program does not run, but might be useful if it could be corrected.
4. An instructor finds a good program, but it runs on another type of computer. The instructor desires to transfer the program to a different make of computer.

### Rules

Before modifying a program, whether it is a simple one with 10 lines of programming code or a more complex one with several pages, consider some fundamental rules.

### Understand What the Program Does (Or What It Is Supposed to Do)

If a program runs and is not very complex, then the purpose of the program is usually obvious. It is useful, especially with more complex programs, for instructors to make a rough diagram, or *program logic chart*, to show what the program does from the standpoint of the student (i.e., what the learner will see on the computer screen). Teachers should run the program, stopping at each logical place where the computer stops, and diagram what happens on the screen. When the program gives two choices, or branches, teachers should illustrate the branch on the diagram and run through each branch on the computer continuing the program logic chart. This process provides teachers with a good idea of what the program does and may suggest modifications which are needed.

### Make One Modification at a Time

It is tempting to make several changes at the same time in the program code when the changes seem so obvious and time can be saved. However, teachers should be cautioned against this practice. One of the changes may affect another part of the program or even many different areas in the program. Instructors should make one modification or change at a time because they will then have a much better opportunity for observing the impact which each change may have on the total program.

### Document Changes

Internal documentation includes the REMark statement which the computer ignores as part of the program listing. Instructors should use REM statement liberally to explain and track their programming efforts. These statements should be made to stand out in the program listing by using a series of asterisks (*****) or other characters which will call attention to the statement as the instructor reads the program code.

At this stage of the programming process, external documentation (information which may help one understand the program) may simply be on scratch notes and output from sample runs. Instructors should always date, number, and sequentially order in folders sample runs so as to trace through modifications if necessary.

### Keep Old Program Listings

If instructors attempt to modify more complex programs, it is possible to get completely lost as to what they are attempting to do; this is especially true when instructors perceive themselves as "advanced" and start trying to move ahead quickly by making multiple changes.

Instructors should always keep a source code listing of the original program and make new listings each time they have made major modifications or a number of minor ones. How often should a new listing be saved?

Instructors can answer that question with another one: "How much of your time and effort are you willing to spend if your diskette is lost or ruined?" *Instructors should make it a practice to date and number each program listing.*

### It Always Takes Longer Than You Expected to Modify a Program

Minor program changes often become major ones. Major program modifications may take longer than it would take to write a new program. Vocational teachers should consider how much the program is worth in its current state and the potential worth of the program if modified. Considerations such as time, new learning opportunities, frustrations, challenges, joys of success, peer recognition, and new teaching aids should be a part of the instructor's judgement regarding whether a program should be modified.

### Modifying Sample Programs

With these rules in mind, instructors can examine some programs and consider how they might be modified.

```
LIST

 2      HOME

 4      PRINT "******** LADDER ANGLE SAFETY ********"

 5      PRINT : PRINT

10     LET C = 3.14159 / 180

20     LET L = 36

30     PRINT "LADDED BASE MUST NOT BE OUT FROM THE HOUSE IN FEET:"

35     PRINT

40     PRINT "NOT LESS THAN-NOT MORE THAN-LADDER LENGTH"

45     FOR L = 36 TO 20 STEP - 2

50     PRINT INT (L * SIN (C * 14) * 100 + .5) / 100, INT (L * SIN (C *

15

          ) * 100 + .5) / 100,

51      PRINT L

55     NEXT L

60     GOTO 327 67

327 67      END

]
```

The preceding program listing, or source code, is written in the computer language BASIC. The program is rather easy to understand with only seven statements or commands. Very long BASIC programs often use only 12-15 different statements, suggesting that they are not as complex as they may initially appear. Some computers accept or understand both upper and lower case letters. This example uses only upper case letters.

Instructors should read a program listing just as they would read a book: one line at a time in sequence. A program listing differs from a book due to the possibility of it jumping forward or backward from a line of code. In a way, this is similar to a programmed instruction booklet. The computer "reads" the BASIC program listing in order, just as ones reads the program.

The numbers on the left are **line numbers**. In BASIC, each line of code must have its own unique number in sequential order. The numbers in this list seem to have been chosen rather randomly and it is difficult to determine if some lines of code have been left out or deleted. The code could be numbered sequentially 1, 2, 3, . . . n, but that would not allow room for modifications. A common numbering sequence is 10, 20, 30, . . . n, which allows room for inserting additional statements. Most computers have small helper programs, or utilities, called RESEQuence or RENUMber, which automatically renumber the lines of code and make other modifications.

Next to the line number is a **command** or a word which tells the computer what it is to do next. The first command at line 2 is HOME. This command tells the **cursor** (a small blinking light on the monitor screen) to go to the upper left hand corner of the screen and also commands the computer to clear or erase the screen. The HOME statement will create a clean work space on the screen.

The PRINT command will print on the screen or on paper anything that is put between the quotation marks ("). Some computers require **both** the opening and the closing quotation marks, while others require only the first set. On more complex programs, one will need to learn how to format, or place the information within the quotation marks, in specific locations on the screen.

Line 5 may seem confusing, as though the computer stutters. There is nothing behind the PRINT statement and no quotation marks. There is a simple explanation for this line. The PRINT command is a simple way of leaving a blank line on the screen or on a printout. The colon (:) tells the computer that there is more than one command on line 5, so the programmer has told the computer to save two blank lines between the title and the next feature. This will be seen on the terminal at line 30 of the listing.

The third statement is LET at line 10. This is a simple method for assigning or changing one entry to another so that it may be used later more easily by the programmer and the computer. Line 20 means that any time L is used in the program, it represents or is equal to the number 36 and can be used in calculations, as in lines 45 and 50, or to print a result, as in line 51.

The first change instructors may wish to make is at line 30. Readers have learned three command statements, HOME, PRINT, and LET, which make up most of this program. In fact, these are common commands which will be seen in most BASIC program listings.

The commands at line 45 and 55 work together as a pair. The FOR command instructs the computer to begin a sequence of activities, or a loop, and the NEXT command informs the computer to continue this loop. The loop includes lines 45, 50, and 51. The command NEXT L at line 55 instructs the computer to go back and re-execute lines 45, 50, and 51.

If instructors were to run a printout of this program, they would see the results of the calculations done by lines 45 and 50. Can you discover where the three columns came from?

Line 60 is an unnecessary GOTO statement which tells the computer to go to line 32767. This takes the reader to the last of the seven new commands of this unit, the END statement. Why is the GOTO statement unnecessary? Not all computers require an END statement, but it is a good idea to state exactly when one has reached the end of the program. The END command may be located anywhere in the program and does not necessarily have to be at the end of the program listing.

**Improving a Program**

It is up to teachers to find programs which run, but which they would like to expand, improve, or customize for specific teaching situations. For example, what does the next program, RECIPE COST, do? How could the program be improved? Could it be changed to find other costs?

```
]RUN
RECIPE COST

NUMBER OF INGREDIENTS?2
INGREDIENT 1:
     COST FOR BULK UNIT IN STORE? 50
     NUMBER OF UNITS IN BULK? 2
     NUMBER OF RECIPE UNITS PER BULK UNIT? 25
     NUMBER OF RECIPE UNITS CALLED FOR? 1
INGREDIENT 2:
     COST FOR BULK UNIT IN STORE? 50
     NUMBER OF UNITS IN BULK? 40
     NUMBER OF RECIPE UNITS PER BULK UNIT? 1
     NUMBER OF RECIPE UNITS CALLED FOR? 1
NUMBER OF SERVINGS? 1

TOTAL COST FOR 1 RECIPE? $6.25
```

```
CHANGE NUMBER OF SERVINGS (1 = YES, 0 = NO)? 0

]

LIST
10     PRINT "RECIPE COST"
20     PRINT
30     PRINT "NUMBER OF INGREDIENTS";
40     INPUT N
50     FOR I = 1 TO N
60     PRINT "INGREDIENT"; I;":"
70     PRINT "COST FOR BULK UNIT IN STORE";
80     INPUT C
90     PRINT "NUMBER OF UNITS IN BULK";
100    INPUT U
110    PRINT "NUMBER OF RECIPE UNITS PER BULK UNIT";
120    INPUT F
130    PRINT "NUMBER OF RECIPE UNITS CALLED FOR";
140    INPUT R
150    P = P + C / U / F * R
160    NEXT I
170    PRINT "NUMBER OF SERVINGS";
180    INPUT S
190    PRINT
200    PRINT "TOTAL COST FOR 1 RECIPE = $"; INT (P * 100 + .5) / 100
220    PRINT
230    PRINT "CHANGE NUMBER OF SERVINGS (1 = YES, 0 = NO)";
240    INPUT N
250    IF N = 1 THEN 170

]
```

Examining the program listing for RECIPE COST, you see that there are only two new commands: INPUT and IF. The computer will wait patiently for the user to input or type in information which the computer needs to do its job. Line 30 asks the user to type in the number of ingredients. The computer will store the information given, such as the number 5, in a location in the computer memory which the programmer has named N. At line 80 the cost for bulk unit in store has been named C.

Notice line 150. The letter P equals or stands for the calculation which follows it on that line. In the first program, the programmer would have said LET P = ....; this illustrates that the optional LET statement can increase the readability of the program listing.

At line 250, the IF statement provides the computer with a way of making a decision. Line 230 asks whether to change the number of servings by answering 1 for YES and 0 (zero) for NO. If the answer is yes (1) the computer "jumps" to line 170 and runs again.

There is a N at line 40 and another one at line 240. Are these the same number? Could these numbers be mixed up in the program? What if the programmer lost track of where and how the letter N was being used? A **very useful** command which has not been used in these programs is the REMark statement. If the programmer had written

```
2 REM   N = number of ingredients
4 REM   C = cost of bulk unit store
6 REM   S = number of servings
8 REM   N = change number of servings
```

He/she might not have used the letter N a second time. REMark statements are ignored by the computer, which allows the programmer to interject remarks or reminders in the program listing without changing in any way what the program executes or does. REM statements take additional room or memory in the computer, but they are important for improved readability or internal documentation.

**A Bigger Challenge**

By this point in this unit, readers have had opportunities to look at a small program, find an error, and correct it. Programs have been presented which run satisfactorily, but can be improved for specific teaching situations with better on-screen instructions, improved formatting, or greater complexity. Since these tasks can be accomplished fairly easily, readers are encouraged to learn more BASIC, either through a workshop or by using one of the many manuals which are available.

The next two reasons for modifying software may lead vocational teachers to spend an unexpected amount of time and effort for which they may not be prepared. With fair warning given, readers should consider the potential success which may result. These challenges for the intermediate programmer are modifying a program which doesn't run and modifying a program which runs on one brand of computer, often from a listing in a book or magazine, so that it will run on a different brand of computer.

One of the greatest challenges to the programmer is a program which does not run, but looks interesting. If the program listing is rather long, perhaps several pages, and is poorly documented without REM statements, analyzing the intended purpose of the program can be comparable to solving an intricate puzzle. Perhaps this explains the fascination some people find in programming; it may be an interesting feature for readers of this handbook.

The following is a more complex program listing for the testing of woodworking knowledge. Other than a DIMension statement which saves some storage space in the computer, a CALL statement which obtains information from the computer, and TAB and CTAB which control the horizontal and vertical tabs in a manner similar to that of the tab key on a typewriter, there are no new commands. If the program is entered correctly, it will run. Perhaps this will give readers some idea of how difficult it can be to find only one error which might cause a program not to run.

```
>LIST
  100  R = O
  110  DIM N$(15)
  120  CALL -936
  130  VTAB 10
  140  INPUT "WHAT IS YOUR NAME?", N$
  150  CALL -936
  160  VTAB 5
  170  TAB 14
  180  PRINT "HELLO,"; N$; "!!!"
  190  FOR I = 1 TO 1500
  200  NEXT I
  210  VTAB 10
  220  PRINT "THIS IS A TEST OF YOUR KNOWLEDGE OF THE"
  230  PRINT "REQUIRED WOODWORKING PROJECT."
  240  PRINT "THIS IS A MULTIPLE CHOICE QUIZ. YOU"
  250  PRINT "WILL BE GIVEN FOUR QUESTIONS."
  260  PRINT "TYPE A 1, 2, 3, OR 4 AFTER THE"
  270  PRINT "QUESTION MARK"
  280  PRINT: PRINT "GOOD LUCK," ;N$"!!!": PRINT : PRINT
  290  PRINT "WHAT TYPE OF JOINT IS USED FOR THE LEG?" : PRINT
  300  PRINT "1) DADO 2) LAP 3) BUTT 4) MORTISE AND TENON";
  310  INPUT A: IF A = 1 THEN 350
  320  PRINT
```

```
330  PRINT "SORRY . . . . IT IS A DADO JOINT. YOU SHOULD"
340  PRINT "HAVE ANSWERED WITH A 1.": GOTO 380
350  PRINT
360  PRINT "VERY GOOD, ";N$;"! HERE'S ANOTHER."
370  R = R + 1
380  FOR I = 1 TO 3000: NEXT I
390  CALL -936: VTAB 12
400  PRINT "WHAT TYPE OF SCREW IS USED IN THE"
410  "BOOKSTAND?": PRINT
420  PRINT "1) ROUND HEAD 2) FLAT HEAD 3) OVAL HEAD"
430  PRINT "4) SCREWMATE";
440  INPUT A: IF A = 2 THEN 460
450  PRINT: PRINT "TOO BAD . . . . IT IS A FLAT HEAD SCREW.": GOTO 490
460  PRINT
470  PRINT "PRETTY GOOD! LET'S DO ANOTHER ONE."
480  R = R + 1
490  FOR I = 1 TO 3000: NEXT I
500  CALL -936: VTAB 12
510  PRINT "WHAT GRIT OR SANDPAPER DID YOU FIRST"
520  PRINT "USE ON YOUR BOOKSTAND?"
530  PRINT: PRINT "1)802)1003)1804)240";
540  INPUT A: IF A = 2 THEN 590
550  PRINT
560  PRINT "NO! YOU SHOULD USE 100 GRIT PAPER"
570  PRINT "FIRST, THEN GO TO A HIGHER GRIT AS THE"
580  PRINT "SURFACE OF THE WOOD GETS SMOOTHER.": GOTO 620
590  PRINT
600  PRINT "RIGHT!! LET'S MOVE ON, ";N$
610  R = R + 1
620  FOR I = 1 TO 3000: NEXT I
630  CALL -936: VTAB 12
640  PRINT "WHAT TYPE OF WOOD IS USED FOR THE"
650  PRINT "BOOKSTAND?": PRINT
660  PRINT "1)WALNUT 2)CHERRY 3)MAHOGANY 4)PINE";
670  INPUT A: IF A = 4 THEN 710
680  PRINT
690  PRINT "ABSOLUTELY NOT! WHITE PINE SHELVING IS"
700  PRINT "USED FOR THE REQUIRED PROJECT.": GOTO 740
710  PRINT
720  PRINT "GOOD GOING, PAL!"
730  R = R + 1
740  PRINT: PRINT
750  IF R = 4 THEN 800
760  IF R = 2 THEN 910
770  PRINT "NOT BAD, BUT YOU NEED TO SPEND A LITTLE"
780  PRINT "TIME STUDYING THE PROJECT PLAN SHEET."
790  END
800  PRINT "WOW!! THAT'S SUPER, ";N$;"!!"
810  PRINT "YOU REALLY KNOW YOUR STUFF!!"
820  PRINT: PRINT "THAT'S ALL WE WILL DO FOR NOW."
830  PRINT "I'LL THINK UP SOME MORE QUESTIONS FOR"
840  PRINT "YOU TO TRY LATER. I HOPE I CAN HELP"
850  PRINT "YOU LEARN MORE ABOUT WOODWORKING."
860  PRINT : PRINT "SEE YA' LATER, PAL!"
880  PRINT "YOUR NEXT QUIZ WILL BE ON CALCULATING"
890  PRINT "THE AGE OF PETRIFIED LOGS (HA! HA! HA!)"
900  END
910  PRINT "YUCHH! THAT WAS DEFINITELY NOT TOO"
920  PRINT "SWIFT! BACK TO THE BOOKS FOR YOU,"
930  PRINT N$;"!!"
```

```
 940  PRINT
 950  PRINT "BETTER LUCK NEXT TIME!"
 960  END
>
```

### A Program Logic Chart

Examine the program listing from the beginning. Begin with line 140, which asks for a name and waits for a response from the Keyboard. At 180, the computer prints HELLO and continues. Begin drawing a program logic chart attempting to predict what one would expect to see on the screen if the program were running. Where does the program branch or change direction? What parts of the program seem to connect in logical ways? Complete a logic chart for the woodworking program and add REM statements which would improve the readability of this program. Sketch a form or template which shows what might be printed on the screen.

Once readers understand what the program is designed to do and how it is programmed to complete the task, they will be better prepared to attempt to get the program to run. Beginning with a logic chart, make an effort to get each logical section or sequence to run on the computer. This cannot be done without some knowledge of programming, but it is a good way to learn. At this stage, one often begins to ask for help from more experienced programmers. A caution, please: don't expect miracles from programmers, even experienced ones. It is almost as difficult for them to understand a program the first time as it is for the instructor. First, display the logic chart and screen template in order to help explain what the program should do. The REM statements should stand out clearly. The programmer might then be able to understand the program logic and see why it will not run.

### BASIC Isn't

The last type of modification, transferring or translating a program listing from one type of computer to another, is also very common. Many magazines, books, and proceedings have useful program listings. Sometimes a program which an instructor would like to use is being used at another school, but runs on a different type of computer.

Transferring a program written in BASIC from one computer to another seems as if it should be easy, but often turns out to be more difficult than expected (see pg. 4). The BASIC language on each brand of computer is far more difficult than one might expect, given the name of the language and the original purpose for which it was developed. Transferring programs is a good way to learn a bit more about programming, usually all the things that instructors would not have done if they had been the original authors.

When instructors begin to transfer BASIC programs from one brand of computer to another, they should obtain one of the useful books listed in the Resource Section of this unit.

## SUMMARY

Program modifications can be quick and easy or take an agonizingly long amount of time. They can be a simple fix or provide an innovative and complex new teaching resource. Vocational teachers can learn something new, have some challenging fun, or be the start of the building for a time. Teachers should be encouraged to try making those little changes which they know will improve a computer program.

## ACHIEVEMENT INDICATORS

1) Explain four major reasons why instructors may need to modify software.
2) Describe five fundamental rules of software modification.
3) Correctly use nine statements of the basic programming language.
4) Modify computer program listings of up to one page which may obtain minor errors or be in need of improvements.

## RESOURCES

Radin, S. & Fayvian, L. (1984). Computers in the classroom—a survival guide for teachers. Chicago: Science Research Associates.

Walter, B. (1984, May). Keeping what works working. *Today's Office*, pp. 23-24.

Watt, P. (1984, June). Software for a donation. *InfoWorld*, pp. 36-38.

**Public Domain Software**
American Software Publishing Co.
1010 Sixteenth Street, NW
Washington, D.C. 20036
202/887-5834

P.J.S. Co.
1062 Taylor St.
Vista, CA 92083
619/727-1015

PC-SIG
1556 Halford Ave.
Suite 130Z
Santa Clara, CA 95051
408/730-9291

## Unit 6

# Designing Software

### UNIT OBJECTIVE

Upon completion of this unit, the learner will be able to apply simple skills for designing educational software. This knowledge will be demonstrated through completion of the achievement indicators at the end of this unit.

### SPECIFIC OBJECTIVES

Upon completion of this unit, the reader will be able to:

1) Define the role of the teacher in the software design process.
2) Define the role of the computer programmer in the software design process.
3) Describe 12 steps in the design process in preparation for the actual coding of a computer program which might be used as a teaching aid.
4) Complete 6 of the 12 steps in the design process in preparation for the actual coding of a computer program which might be used as a teaching aid.

# Designing Software

### BY: DR. ROBERT WATTS

Designing software has two major components: design and software. The first part, design, includes the identification of needs or objectives (what one wants to do) and the detailed ways of reaching those objectives. This part uses the skills of the educational specialist regarding what is to be taught, how it can be effectively taught, and why it is being taught. The second part, software, is computer language coding, which tells the computer what to do and guides the student in the learning process. Coding in various computer languages such as BASIC, LISP, LOGO, PASCAL, or PROLOG, is done by programming specialists who understand data structure, logic, syntax, and efficiency. The two specialists do not necessarily understand each other.

Educational software has been criticized by teachers and students for a variety of reasons (readers are referred to unit C.4, B.5, & E.1.2.3.). Vocational teachers need to examine the underlying problems with early educational software, often written by individuals who knew little about curriculum and instruction or by educators who knew what they wanted to do, but lacked the skills of programming specialists.

The purpose of this section is to provide vocational instructors with skills and ideas for improving educational software. This unit will not prepare vocational teachers to be programming specialists, but it will focus on the tasks needed to help produce good educational software. This is 80% of the process, the area in which teachers are experts in the designing phase prior to coding software in a programming language. This unit will progress to the stage of computer programming. At that stage, instructors and their administrations can decide whether their professional time should be spent as educators or as programmers.

This unit describes design fundamentals and design specifics. Designing a computer program is a logical exercise in problem solving. Instructors might have something they want to do, like teach a skill, give insight into an understanding, or set up a drill-and-practice exercise. In each case, instructors are simply using the computer as an aid or tool to help reach educational objectives. Vocational teachers should consider these problems in a logical way, breaking them into subunits or smaller problems, and then reassembling them into workable systems.

Twelve steps follow, eight of which guide readers through the logical development of an educational software program. A description of, or example for each step is given. Readers should examine the material quickly to get an overview of the process and then go through it again to design their own programs.

1. Completely understand the problem that is to be solved. What do you want to do? Complete an overall program design.

Sample Problem

You want students to learn how to use a simple parts inventory system. Your objective is to complete a computer program that will keep track of a number of different parts in several categories. Data will be entered at the microcomputer and a detailed report about the parts inventory will be produced, upon request. The student will be able to add or delete parts from the inventory.

Sample Program Design

```
                              START (1)

                              INPUT (2)

        Inventory                                        Inventory

                             PROCESS (3)

          Code                                            Record
          File                                             File

                             OUTPUT (4)

                              STOP (5)
```

2. Determine and understand the exact output desired.

Often, the key to a good design is the ability to visualize the exact output, or what you would finally like to see on the computer screen or on a paper printout. Once the output, or final report, has been determined to be correct, the information or input and the processing needed to present the final output can be completed.

Sample Output

The output report, entitled "Parts Inventory," should appear as shown below.

---

```
Parts Inventory
Part          Part           Part          No. on        Original
Number        Name           Category      Hand          Cost

xxxxxx        xxxxxxxxxx     xxxxxxxxx     xxxx          $xxxx.xx
                                           Total Cost    $xxxxx.xx
```

There should be a line containing all details, or a detail line, for each part. The detail line shall include part number, part name, part category, number on hand, and original cost. Total cost of all inventory on hand will be printed.

3. Identify the information sources (where does it come from and how will the student obtain the information) and the type of processing needed.

Sample Input

As each new part is obtained, the student will get the data from the original invoice (or a copy) and enter the data into the microcomputer. Parts that are used must be deleted from the inventory. Students will obtain this information from the requisition form (or a copy). Data will be saved as a file on a floppy diskette, one record for each part. The input data records will have the following field contents:

Part Number:

Part Name:

Part Category:

No. on Hand:

Original cost:

The student will be prompted for input, or asked for information, as follows:

Do you want to add or delete a part? (A = add; D = delete)

Sample Processing

Processing will include adding new parts, deleting parts, and keeping track of the total cost for all parts. The computer will perform processing as needed to produce the desired report (see section 2). The end of the input file will be signified by a part number of 999999. There will never be more than 50 input records for a given category.

4. Develop logical subdivisions for the overall problem.

These subdivisions describe in more detail the overall program design you developed in Step 1.

Sample Logical Subdivisions

— initialize variables, or determine the starting point for each item to be used in the program

— enter the input data, or add/delete parts information, via the micro

— read input detail records until last record (999999)

— look up part category, or code, from diskette for each new input record

— print report heading

— print a detail output line for each input record

— after last input record print total cost of inventory on hand

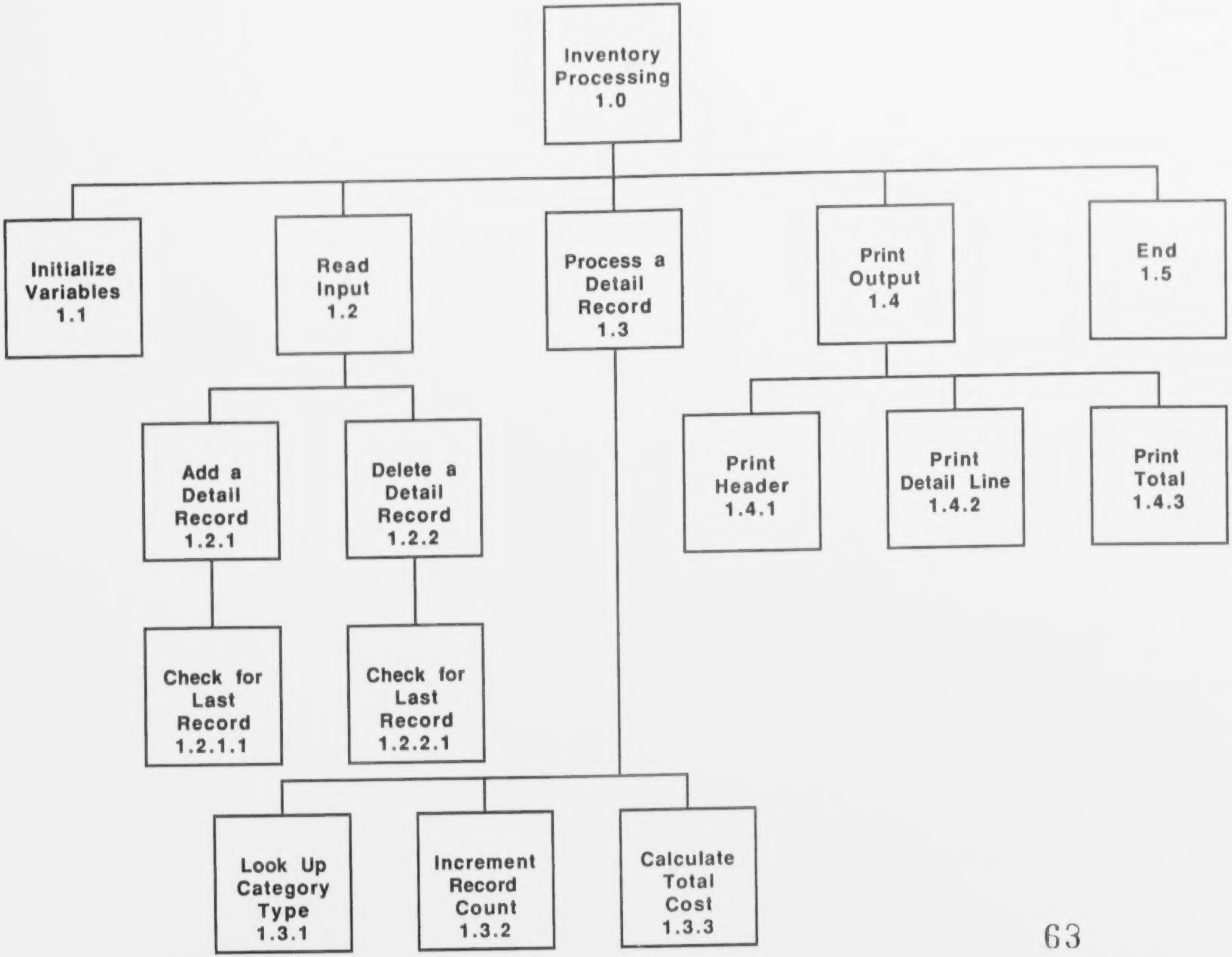
— end

5. Draw a hierarchy chart.

A hierarchy chart is simply another way of integrating and describing what you have done in Steps 1 and 4. Rules for construction of a hierarchy chart are:

1. Each module should be independent of the other modules, i.e., each module should be executed when control is passed to it by the module directly above it.
2. Once a module has been executed, control should be passed back directly to the module above.

Sample Hierarchy Chart


Inventory Processing 1.0
Initialize Variables 1.1
Read Input 1.2
Process a Detail Record 1.3
Print Output 1.4
End 1.5
Add a Detail Record 1.2.1
Delete a Detail Record 1.2.2
Print Header 1.4.1
Print Detail Line 1.4.2
Print Total 1.4.3
Check for Last Record 1.2.1.1
Check for Last Record 1.2.2.1
Look Up Category Type 1.3.1
Increment Record Count 1.3.2
Calculate Total Cost 1.3.3

6. **Develop a detailed program design.**

The detailed program design places the modules you designed in Step 5 into a logical sequence which will eventually be executed by the computer. Remember, you are doing this to help describe to the computer specialists exactly what you want the computer to do for the student.

Sample Detailed Program Design

| | |
|---|---|
| Inventory Processing Main Program | 1.0 |
| Initialize Variables | 1.1 |
| Read Input | 1.2 |
| Add a Detail Record | 1.2.1 |
| Delete a Detail Record | 1.2.2 |
| Check for Last Record | 1.2.1.1<br>1.2.2.1 |
| Process a Detail Record | 1.3 |
| Look Up Category Type | 1.3.1 |
| Increment Record Count | 1.3.2 |
| Calculate Total Cost | 1.3.3 |
| Print Output | 1.4 |
| Print Header | 1.4.1 |
| Print a Detail Line | 1.4.2 |
| Print Total | 1.4.3 |
| End | 1.5 |

7. Write the pseudocode (or false code).

At this point, you are entering the realm of the programmer. But there is no reason why you cannot begin writing out the detailed steps the computer will need to execute or run your program. You are not writing the computer code. Pseudocode provides the opportunity to complete a detailed description of the program without having to worry about the syntax of writing in a computer language. It is a detailed high level cognitive process for planning what you want to do versus the rather mechanical process of coding. A programmer can take the pseudocode and write the actual code in almost any computer language available on the microcomputer.

8. Desk-check the logic in the pseudocode.

Desk-check means to check the pseudocode for logical errors in the logical structure of the program, errors that the educational specialist should be able to identify more often than the computer programmer. Go back through Steps 1 - 6 and be sure the program will do exactly what you want it to do.

You can desk-check your own code by going through the code very carefully, but like trying to proofread your own typing, it is better to have someone familiar with what you are trying to do desk-check your pseudocode.

9. Write the actual program code on paper.

This is often the point at which the layperson tries to begin writing a computer program. Up to this point, the choice of programming language or type of computer has not really been a factor. Through Step 8, enough thought and detail have gone into the planning so that changes are unlikely to occur during the coding process. The results of the working program should be of no surprise to anyone involved in the project.

A competent programmer should be able to complete this program in a few hours.

10. Key in the program.

This is the first time the computer is actually used. Once the program has been written, it must be keyed into the computer. This means simply copying exactly what the programmer has written in Step 9 and saving it on a diskette.

11. Debug the program.

It is not unusual for a program not to run properly the first time, even when the design and coding processes have been done meticulously. It may not run due to typing errors when entering the program, but hopefully not due to an error in program logic. You and the programmer can work together effectively in debugging the program so that it runs to your satisfaction.

12. Document the program.

Every good programmer knows that internal documentation, which is part of the program code, should be included as the program code is being written in Step 9. Once the program is running as anticipated, other types of program documentation should be developed.

External documentation, outside the program code—perhaps in a folder—is for the program user (teacher and/or student). The instructor who uses the program may differ from the one who developed the program. One should assume that this person or the student knows absolutely nothing about using the computer or this particular software.

System documentation should include all the documents that were developed during the design process (Steps 1 - 9) and should be kept in sequence along with the final source code (Steps 10 - 11), or computer language program code, and a successful sample program run that includes the use of addition and deletion of parts. All subsequent changes to the program should be added to the system documentation and all of this kept in a safe place away from students.

## ACHIEVEMENT INDICATORS

1) Define the role of the teacher in the software design process.
2) Define the role of the computer programmer in the software design process.
3) Describe 12 steps which guide one in the software design process.
4) Complete 6 of the 12 steps in the design process in preparation for the actual coding of a computer program which might be used as a teaching aid.

## REFERENCES

Becker, S. (1984, April). The competent course designer. *Data Training*, pp. 24-25.

Billings, K. Software development: A publisher's perspective. *AEDS Monitor, 22* (3), 8-13.

Bonner, P. (1983, August). Computer programming: What's in it for you? *Personal Computing*, pp. 128-137.

Crawford, C. (1983, September). Why you should learn to program. *Popular Computing*, pp. 153-156.

Datiles, U., & Carr, V. (1979, February). A cooperative approach to the development of instructional materials. *T.H.E. Journal*, pp. 40-41.

Debenham, J. (1978). Computer automated simulations: How to design one for under $300. *Educational Technology, 18* (8), 30-33.

Dimas, C. (1978). A strategy for developing CAI. *Educational Technology, 18* (4), 26-29.

Frillman, L. (1982). The development of your own programs for the microcomputer—a process approach. *American Annals of the Deaf, 127*, 625-630.

Isaacson, D. (1984). Designing user friendly software. In R. Lanbert & J. Rodenstein (Eds.) *Microcomputers: Applications in career counseling and career education*, (pp. 97-110). Wisconsin-Madison.

Kearsley, G. (1979). A model for debugging instruction. *Educational Technology, 19(3)*, 50-52.

Kehrberrg, K. (1979). Microcomputer software development: New strategies for a new technology. *AEDS Journal, 13* (Fall), 103-110.

Kontos, G. (1984, May/June). Cost-effective, teacher-made, quality CAI—a fallacy. *Electronic Education*, pp. 23; 35.

Kosel, M., & Jostad, K. (1983, Fall). Designing the display. *The Best of PIPELINE*, pp. 29-32.

Lundeen, R. (1982, December). Improving readability. *Creative Computing*, pp. 254-255.

Mace, S. (1984, April). Computer erector sets: Software's missing link. *InfoWorld*, pp. 38-40.

Van Kampen (1983, October). Notes on human engineering. *inCider*, pp. 76-80.

Zahniser, G., Long, J., & Nasman, L. (1983). *Microcomputers in Voc Ed: A Decision Guide*. Columbus, OH: The National Center for Research in Vocational Education.

## Unit 7

# Prepare Instructional Materials to Accompany Software

### UNIT OBJECTIVE

Upon completion of this unit, the learner will be able to prepare instructional materials to accompany microcomputer software. This knowledge will be demonstrated through completion of the unit achievement indicators.

### SPECIFIC OBJECTIVES

Upon completion of this unit, the learner will be able to:

1) Explain the relationship between instructional materials and microcomputer software.
2) Describe the need for instructional materials in Computer-Based Instruction.
3) Explain considerations which should precede the preparation of instructional materials for Computer-Based Instruction.
4) Discuss characteristics of effective instructional materials.

# Prepare Instructional Materials to Accompany Software

### BY: DR. ROGER A. RANKIN

The inclusion of Computer-Based Instruction (CBI) into vocational curricula and instruction can achieve several outcomes:

- enhance learning experiences for students based on their individual needs.
- stimulate subject matter interest for each student.
- motivate students to utilize capabilities of the microcomputer.
- provide students information relevant to their occupational objectives.
- achieve instructional objectives.

If vocational teachers are limited to teaching with educational objectives, lesson plans, and microcomputers, they will fall short of their instructional goals. Teachers are not perfect; they are not continually stimulating, motivating, relevant, and all things to all students (The Center for Vocational Education, 1977). Most instructors rely on additional help to aid in the teaching process in the form of instructional materials—materials for students to use, read, listen to, or view. Vocational teachers who have developed quality instructional materials can meet the needs of students regardless of their abilities to perform (The Center for Vocational Education, 1977). The trials and tribulations of integrating CBI into vocational curricula and instruction can be considerably eased if instructors develop appropriate information sheets, handouts, and software documentation summaries and exercises.

Instructors will need more than a CBI plan and good intentions to successfully integrate computers into the vocational classroom. Instructional materials will help vocational teachers to reach program goals.

Although a wealth of software has been developed for the educational market, many software packages cannot be directly infused into vocational curricula and instruction. In addition, a majority of the educational software available does not meet the specific needs of vocational education. This lack of software for vocational education users, coupled with software accompanied by poor documentation, poses a definite problem for instructors teaching with microcomputers. To address this problem with the attitude, "Oh well, I'll just wait until the right software comes along," may be to wait a long time. Instructors must face up to the current status of software and then prepare instructional materials that will make up for the shortcomings in software and related documentation.

Another consideration stems from the laboratory setting in which students interact with microcomputers. This environment can render vocational instructors grossly ineffective if good instructional materials are not in place. Instructional materials are needed which will lead students through the software step by step, with examples and a system for evaluation. Frequently, especially in a microcomputer laboratory, vocational teachers will have to spend a lengthy period of time with a student. During this period of time, other students may need similar types of help. In this case, instructional materials can pay off the most. Good instructional materials can lead students through difficult segments of curricula. When necessary, teachers can refer to the specific materials to get students on the right track. The example "NOW, YOU'RE ON YOUR OWN!" (see Appendix) shows how a worksheet can be developed to assist students through a programming exercise. A teacher can have a large number of students writing this basic program and at the same time answering individual questions by referring students to a specific item on the handout. Without this type of handout, teachers would spend lengthy periods of time with each student as he or she encountered a problem. Frequently, students will encounter the same learning difficulties at different times in the class period according to their individual progress. There is no reason for instructors to repeat instruction or show students the same thing time after time when an appropriate handout will do the job. Handouts, information sheets, job sheets, examples, and worksheets will meet these needs as well as tutorial software, film strips, and other traditional instructional materials.

As instructors teach students via microcomputers, they must realize that the typical day is made up of blocks of time and that they must work within those time constraints. Therefore, instructional materials must be designed with the appropriate timeline in mind. For example, it would not be wise for the business education teacher to give students a word processing software package and a large book of documentation at the beginning of a 50 minute class period and say , "Have at it, we'll continue tomorrow where you leave off today." This "whole main course" approach to teaching implies that students must "inhale the meal" as opposed to eating it bite by bite. A logical alternative would be the development of information sheets designed to lead students to a specific point at the end of a 50 minute class period. These instructional materials should be available to students during each laboratory session.

The lengthy documentation that accompanies many software packages are difficult to understand and digest. This situation requires teachers to modify documentation and present it to students in a logical manner. An example of this process (*e.g., WordStar Handouts) is found in the Appendix at the end of the unit. This example shows what a business education teacher could provide students at the beginning of a unit which features WordStar word processing software on the IBM Personal Computer. A quick summary of the documentation, in information sheet form, can lead students through software step by step. This example, the first handout, "GETTING STARTED," is designed to lead students through the simple path of creating a document with WordStar, saving it, and printing it. This exercise which allows a brief introduction to the software, is designed to take approximately 45 minutes, leading the novice through a very elementary exercise that gives an overview of what the word processing package will do and how it works.

The second handout, "FREQUENTLY USED WORDSTAR COMMANDS," summarizes important commands students will use while learning the software package. This handout can serve as a ready reference for students. It can save students valuable time wasted in searching through documentation or repeatedly asking teachers for information.

The WordStar handout, "MENU MAP," depicts various menu options available to assist students. This handout allows students to gain assistance easily and individually.

Additional examples of instructional materials in the Appendix are titled "COMMANDS THE APPLE UNDERSTANDS" and "TRS-80 GRAPHICS." While the "COMMANDS" handout is not specific to any software package, it is a summary of commands, statements, and thoughts that can be given to students to assist them in using specific hardware as well as programming on that hardware. Students will find that they will need these commands to interact with software programs as well. The "TRS-80 GRAPHICS" handout is an excellent example of one which introduces new information to students, gives examples, and assigns a work activity to be conducted with information on the handout.

Unit C.3 discusses lesson plans with regard to implementing CBI. It is this plan, when accurate, that helps instructors identify important considerations before they prepare instructional materials. The identification of subject matter content, goals, and objectives will help instructors determine what students must have in their hands to master assignments. The individual needs and learning styles of students are prime considerations for teachers in this process of developing instructional materials. Some students may perform well starting with an information sheet and software, while other students need an introduction from the teacher and step by step directions for the first few class periods. Often, students are able to work in small groups helping each other through spots and relying on the appropriate instructional materials.

A key consideration in the preparation of instructional materials is the reading level of students. A technically written explanation of how to perform a task will not help a student with a low reading level. However, alternative methods of explanations may permit that student to perform the task in a very capable manner (The Center for Vocational Education, 1977). Instructors must evaluate the variety of mechanisms available for determining reading level and work closely with counselors and other appropriate staff to ensure that instructional materials are effectively used by students.

The design of quality instructional materials requires instructors to implement a well-defined plan. Instructional materials should be designed to lead students through activities which accomplish the instructional objectives of the vocational program. Instructional materials should be "user friendly," not vague, not assume too much of students, and be at the correct reading level. Quality instructional materials can guide students through the difficult learning spots of Computer-Based Instruction.

## SUMMARY

Instructors who implement CBI into vocational curricula should recognize that software development is at the beginning stage. Consequently, vocational teachers will need to prepare instructional materials to accompany software and its respective documentation.

Instructional materials should occur with grade levels, reading levels, needs of students, and instructional objectives. Daily lesson plans can help instructors identify the types of instructional materials needed for specific instructional activities.

Instructional materials should lead students a step at a time; they should not be constructed with the assumption that students will grasp the entire software program at one time. Students interact with microcomputers for specific time periods during the school day. Instructional materials should be constructed in accordance with student hours for microcomputer use.

Instructional materials can be in the form of film strips, cassettes, information sheets, handouts, and examples. Individual handouts and information sheets that guide students through software are very effective. These types of instructional materials summarize or emphasize specific portions of software documentation.

Vocational teachers cannot wait until publishers provide software and documentation to meet the needs of vocational education students. Instructors should recognize the shortcomings of software and prepare instructional materials that can guide students through difficulties they may encounter with Computer-Based Instruction.

## ACHIEVEMENT INDICATORS

1) Explain the relationship between instructional materials and microcomputer software.
2) Describe the need for instructional materials in Computer-Based Instruction.
3) Explain considerations which should precede the preparation of instructional materials for Computer-Based Instruction.
4) Discuss the characteristics of effective instructional material.

# APPENDIX

## NOW, YOU'RE ON YOUR OWN!

Objectives

Write a program that presents a story, asks four questions, grades the student's response, and gives the student a percentage score.

Some Hints and Help —

You will need to:

Statement
Needed

1) Make it friendly .......... P
I
2) Give some directions .......... P
3) Clear the screen after the directions .......... P
I

Example NO 1

```
25 PRINT "HIT RETURN TO CONTINUE"
30 INPUT B$
35 HOME
```

4) Present the story .......... P
5) Clear the screen after the story. Example NO 1 again .......... P
I
6) Ask the first question, accept the student's answer, and grade it
P I

EXAMPLE NO 2

```
60 PRINT "GEORGE WASHINGTON WAS THE"
65 PRINT "A. FOURTH PRESIDENT OF THE UNITED STATES"
70 PRINT "B. THE SIXTH PRESIDENT OF THE UNITED STATES"
75 PRINT "C. THE FIRST PRESIDENT OF THE UNITED STATES"
80 PRINT "D. THE SECOND PRESIDENT OF THE UNITED STATES"
85 INPUT C$
90 IF C$ = "C" THEN X = 1
```

7) Clear the screen after each question—Example NO 1 again
8) Ask the second (and remaining) questions.
Same as Example NO 2 except: X = X + 1 for the **IF** statement
9) Figure students grade and report it.
10) Give students a good-bye message.

(Those individuals who care to, can have a student with a low score reread the story and take the test again—how would you do it??)

## GETTING STARTED . . .

The steps below will guide you through the simple path of creating a document in WordStar, saving it, and printing it.

1. Entering WordStar

At the system prompt

| TYPE: | SEE: |
|---|---|
| WS [RETURN] | OPENING MENU |

2. Opening a Document

At the Opening Menu

| TYPE: | SEE: |
|---|---|
| D | D |
| filename [RETURN] | Name of file to edit? |

3. Entering Text and Saving Your Work

At the Main Menu

| TYPE: | SEE: |
|---|---|
| KD | WAIT<br>saving file: a: filename |

4. Printing Your Document

At The Opening Menu

| TYPE: | SEE: |
|---|---|
| P | P not editing<br>name of file to print |
| filename [ESCape] | |

5. Leaving WordStar

At the Opening Menu

| TYPE: | SEE: |
|---|---|
| X | A > |

## FREQUENTLY USED WORDSTAR COMMANDS

Moving Right and Left:

| | |
|---|---|
| ^D | Moves cursor right one character |
| ^F | Moves cursor right one word |
| ^QD | Moves cursor to right end of current line |
| ^S | Moves cursor left one character |
| ^H | Moves cursor left one character |
| ^A | Moves cursor left one word |
| ^QS | Moves cursor to beginning of current line |

Moving Up and Down:

| | |
|---|---|
| ^E | Moves cursor up one line |
| ^DE | Moves cursor up to top of screen |
| ^W | Leaves cursor in same position; new line appears at top of screen and window on text moves down one line |
| ^R | Leaves cursor in same position; previous screen reappears |
| ^X | Moves cursor down one line |
| ^QX | Moves cursor down to bottom of screen |
| ^Z | Leaves cursor in same position; new line appears at bottom of screen, and window on text moves up one line |

| | |
|---|---|
| ^C | Leaves cursor in same position; next screen appears |
| ^QC | Moves cursor forward to end of file |

Repeating Commands:

| | |
|---|---|
| ^QQ^ | Repeats any cursor movement or scrolling command continuously until stopped |

Deleting Text:

| | |
|---|---|
| ^G | Deletes character at cursor position |
| ^T | Deletes word from cursor position to the right |
| ^Y | Deletes entire line in which cursor is located |
| ^QY | Deletes all characters from cursor position to the right end of the same line |
| DELETE | Deletes character to the left of cursor position |

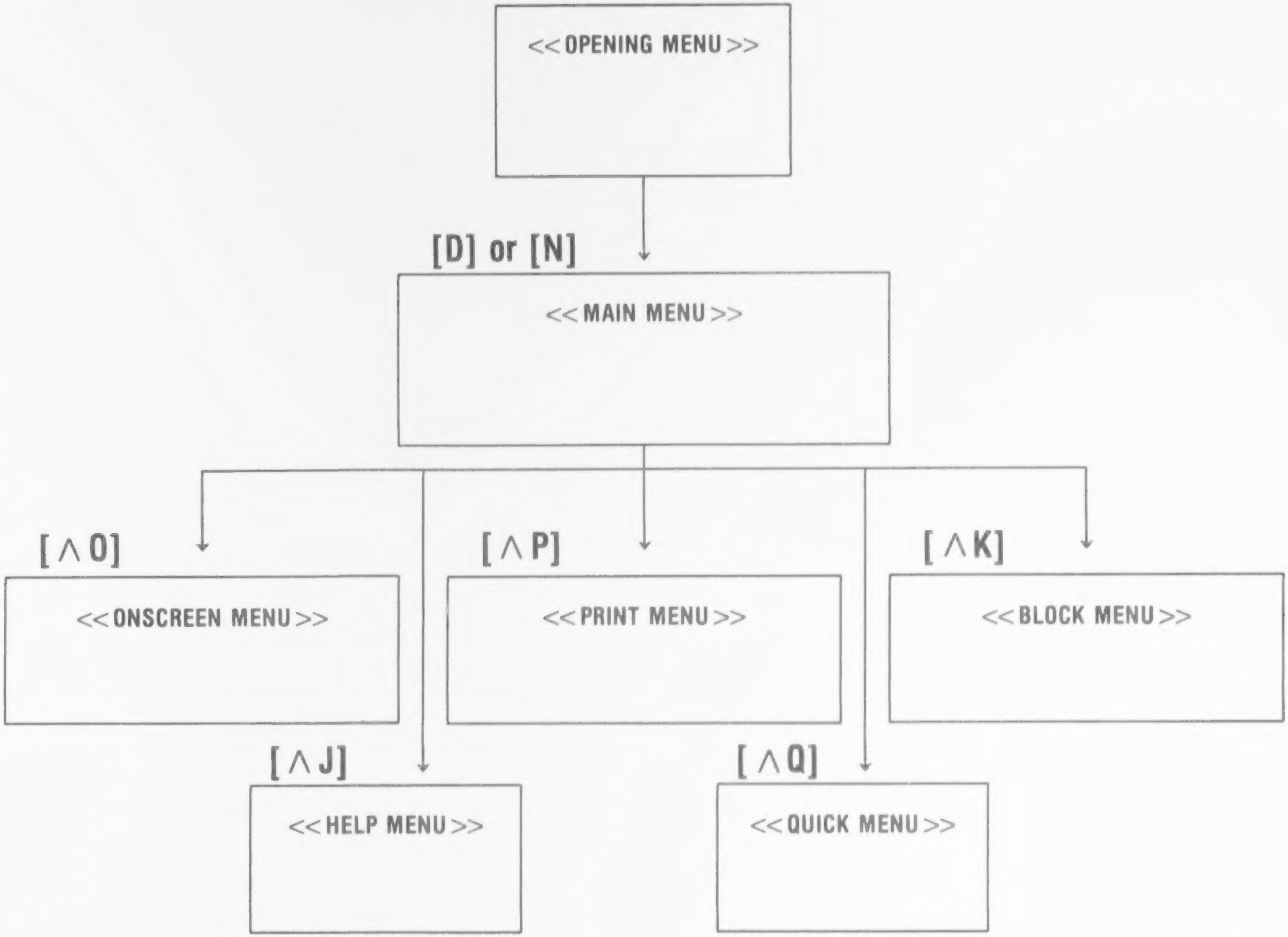
Inserting Text:

| | |
|---|---|
| RETURN | Inserts a hard carriage return, leaving cursor at beginning of previous line (insertion: on) |
| ^N | Inserts a hard carriage return, leaving cursor at beginning of newly inserted blank line |

Reforming paragraphs:

| | |
|---|---|
| ^B | Re-forms paragraphs between current margins |

# MENU MAP



# COMMANDS

## THE APPLE UNDERSTANDS

CATALOG — Will list on the monitor the program on the disk.

HOME — Will clear the screen on the monitor.

NEW — Will erase (or clear) the memory of the computer

RUN — Will run the program currently in the memory of the computer.

RUN (PROGRAM NAME) — Will tell the computer to go to the disk, load a program into the memory, and run it.

INIT (PROGRAM NAME) — Will initialize the disk (get it ready to speak to the Apple) and save that specific program in the disk (Usually called the HELLO program).

LOAD (PROGRAM NAME) — Will tell the computer to load a program from the disk into the memory of the computer, but will not put it on the monitor screen.

LIST — Will list the program in the memory of the computer.

SAVE — Will tell the computer to store what's in the memory in the disk in the disk drive.

# STATEMENTS

## TO BE USED IN BASIC PROGRAMMING

PRINT — Will tell the computer to print everything inside the "quotation" marks.

INPUT — Will tell the computer to stop and allow the user to input data.

IN/THEN — Will tell the computer to compare data and then decide where to go in the program.

GOTO — Will tell the computer to go to a different line number within the program.

# THOUGHTS

## ABOUT PROGRAMMING AND USING THE COMPUTER

You need to be able to play three roles: A programmer
A computer
A user

When writing programs, think like a programmer in order to get the correct program, think like the computer so you can visualize what the program is asking the computer to do, and think like the user so you'll know what the program wants the user to do.

# TRS-80 GRAPHICS

Often, in software programs, you will see screen displays using borders and vertical and horizontal lines. There are a number of ways to do this in BASIC programming, but the command we are going to use is SET.

SET is a command which lights up a point on the screen.

The screen is divided into a matrix with 127 positions across the screen and 47 positions down the screen. The horizontal rows are designated by the letter X, while the vertical columns are designated by the letter Y.

Keeping this in mind, X can equal 0 - 127 and Y can equal 0 - 47.

0 1 2 3 * * *

0

1

2

3

*

*

*

To turn on a point, the syntax used is:

SET (X,Y)

SET can be used in the immediate mode or can be used in program lines.

Let's try a few examples.

Clear your screen.

Type in: SET (63,23)

This lights up a point at position 63 (X) and 23 (Y). Try a few more on your own.

A short program which randomly sets points is shown below. Type in the program to see what it does. (The RND command generates random numbers between ) and 127 for X, and between ) and 47 for Y)

```
10 CLS
20 SET (RND (127), RND (47))
30 GOTO 10
```

Experiment with turning on various points or creating a design using SET in a program.

Used in the programming mode with a few other BASIC statements, SET can be used tc draw horizontal and vertical lines. The following simple programs demonstrate this.

Horizontal line:

```
10 CLS
20 FOR X = 10 TO 100 — tells the computer to start at position 10 and stop
                       at position 100 on the X axis
30 SET (X, 15) — sets X at position 10 to 100 (as defined in line 20),
                 and Y at position 15
40 NEXT X — returns the program to line 20 and repeats until X = 100
```

Type in this program and see how it works. Now write your own program drawing a horizontal line (or several). Use your imagination.

Vertical line:

Vertical lines are programmed in the same manner, but with Y being used in the FOR NEXT statement instead of X. Here's an example:

```
10 CLS
20 FOR Y = 15 TO 45
30 SET (10,Y)
40 NEXT Y
```

Experimenting with vertical lines.

Borders:

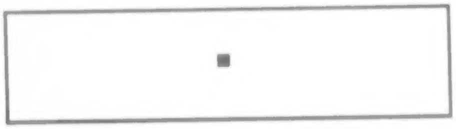
Borders are made by connecting a horizontal line with a vertical line. The sample program will draw a border by combining the two sample programs used to draw the horizontal and vertical lines above.

```
1- CLS
20 FOR X = 10 TO 100
30 SET (X, 15)
40 SET (X, 45)
```

```
50 NEXT X
60 FOR Y = 15 TO 45
70 SET (10,Y)
80 SET (100,Y)
90 NEXT Y
100 GOTO 100
```

Type in this program to see how it works.

Now it's your turn! Draw a rectangle made up of horizontal lines 20 units in length and a vertical line 5 units in length. And while we're at it, let's put a point in the center. Raise your hand if you need help.



ASSIGNMENT: Using the information we have provided, along with the graph paper provided, plan and develop a picture, design, etc., using SET and other BASIC commands. BE CREATIVE . . . (an idea would be to make a chart to visually display students' grades).

Make sure you draw out your figure and write out each statement **before** you begin.

## REFERENCES

The Center for Vocational Education (1977). *Module B-5 select student instructional materials.* Athens, GA: American Association for Vocational Instructional Materials.

The Center for Vocational Education (1977). *Module B-6 prepare teacher-made instructional materials.* Athens, GA: American Association for Vocational Instructional Materials.

## Unit 8

# Modify Software Documentation For Specific Instructional Use

## UNIT OBJECTIVE

Upon completion of this unit, the learner will be able to modify software documentation for specific instructional purposes. This knowledge will be demonstrated by completing the achievement indicators at the conclusion of this unit.

## SPECIFIC OBJECTIVES

Upon completion of this unit, the learner will:

1) Explain the term "on-line documentation."
2) Describe two types of software documentation.
3) State four common problems associated with software documentation.
4) List four writing principles that should be used to improve documentation.
5) Discuss the importance of face validity for software documentation.
6) Develop a checklist of information to be included in modified documentation.
7) Describe the process of field testing modified documentation.

---

Contents of this unit are reprinted with permission, copyright (c) (1984) by Educational Technology Publications Inc.

# Retreading Misaligned Documentation

### BY: MICHAEL T. DROTTER

*Most of us have had to face this decision at least once in our lives. The old car is in pretty rough shape. The front end is so far out of alignment that it has prematurely worn the tires to a frazzle. It is cheaper to buy a set of retreaded tires than to get the front end aligned. However, with the bad front end those new tires won't last long. It is a dilemma; but one thing is for sure, as long as the front end is not performing right—neither will the tires.*

This anecdote is comparable to the dilemma teachers encounter with documentation that inadequately serves instructional needs. Incomplete, misleading, and ambiguous software manuals can stifle the implementation stage of Computer-Based Instruction (CBI). Instructors may find themselves in the position of having to "retread" software documentation to meet teaching needs.

Trainers will encounter many software packages that are accompanied by learning guides of questionable quality (Huntington, 1983). Some program developers seem to be unaware of the significance of documentation. Often, software developers view the tasks associated with documentation as the drudgery end of the program development process (Johnson, 1983).

An increasing number of software publishers are sensing that computer programmers may not be the best writers for software manuals. Creating lines of code does not automatically translate into well written instructional materials. Writers with such diverse backgrounds as education and liberal arts are being sought to assist in the development of documentation (Fawcette, 1983).

The trend among progressive programming organizations is to provide as much on-line documentation as possible. On-line documentation is documentation which appears on the screen during program execution. It is implemented as a "help function" for the user in many software packages. This documentation may also include on-line prompts during the learning session. Quality software will allow the user to choose among several levels of assistance. The utopian goal for training curricula is to include software that is completely self-documenting. Memory capabilities and execution speeds of most microcomputers prohibit the complete internal documentation of software. What this fact means to instructors is that for the near future most software, no matter how sophisticated, is only as good as its printed documentation (Lobello, 1982).

## WHICH DOCUMENTATION CAN BE DOCUMENTED?

Two types of documentation exist: Written manuals and internal program documentation. A starting point for instructors is to determine which internal documentation can be easily modified and which cannot be modified. Special purpose commercial software packages perform functions well, but most cannot be altered by the user. Many software packages reside on specially prepared diskettes that preclude anyone from modifying or copying the program code. Documentation changes for this type of software are limited to clarification of the documentation supplied with the software. For this reason, instructors will usually be limited to modifying written documentation.

If learners are having difficulty using software and its respective learning guide, a complete revision of the manufacturer's documentation may be necessary to create an educational computing document that will really serve learners' needs. Generally, it is easier to initially provide good documentation than to fix poor documentation. This fact makes CBI software that is derived from general software packages (such as spreadsheets or database management software) the easiest to tailor to specific educational purposes. Because specific CBI applications are created within the structure of the general software, documentation can also be tailored specifically to the learner's level of understanding. This documentation should include information necessary for the use of the software package integrated with information relevant to specific instructional applications. Approaching documentation in this manner should increase learner efficiency. Most users prefer to concentrate on the task at hand as opposed to reading several manuals on how to run software (Baldridge, 1983). This integrated method instructs the user about operating software without detracting from the objectives of CBI.

### Common Problems With Documentation

What makes software documentation so difficult to use? Many times, documentation supplied with commercial software is filled with specialized vocabulary and acronyms. Additionally, documentation may require a higher level of understanding of general computing principles than most users possess, contain error messages that can stump a seasoned cryptographer, or omit information that is appropriate to educational computing. Many problems associated with documentation stem from poor technical writing; some forethought by educators will prevent these problems. Four writing principles can be implemented by teachers as they work to improve documentation (Wiio, 1982):

1. use short, simply constructed sentences;
2. use vocabulary that consists of short familiar words;
3. structure the presentation using concrete ideas that can be perceived by the reader; and
4. present subject matter that is pertinent to the learning needs of the reader.

Teachers who improve documentation still risk the injustice that the manual will merely show the software as it is—no simpler and no friendlier. Good quality documentation cannot change poor quality software. To add to that injustice, every mistake that instructors make in the re-writing process can detract from the usability of the software. Documentation writers cannot make the software better, but they can certainly make it worse (Weiss, 1984).

**Creating Usable Documentation**

The first impressions that people form are often lasting impressions. Educators who develop documentation to meet specific training needs must strive for face validity with the manual. Face validity means that the documentation has a professionally prepared appearance. If the documentation is sloppily prepared and ugly in format, or if it looks like a work of technical drudgery, learners will hesitate to follow its intended use (Sohr, 1983). The appearance of the manual should enhance its credibility with the user.

An initial writing step for instructors is to state and define the audience for the documentation (Sohr, 1983; Weal, 1983). It is frustrating for readers to scan well written documentation that is too technical or too simple because the writer failed to define the audience in the preface of the manual.

Educators should organize the manual by task, not by command. For example, topics should be used that explain how to sort data, calculate new values, or print a report; not how to use isolated commands needed to execute a procedure within that procedure (Grimm, 1982). Sohr (1982) suggests moving the explanation from the overall general functions of the procedure through the specific steps of execution.

When instructors modify documentation, they should avoid the use of computer terms and computer acronyms (Grimm, 1982). Computer terms that are included in documentation should be defined within the text and in a glossary of terms at the end of the documentation manual. Documentation should contain complete explanations of error codes generated by the software. Teachers should use examples where possible to clarify and enhance the written text. Logic diagrams should be provided to aid advanced users. Instructors can integrate the text with the actual software to support and augment on-line documentation. Advanced users may resist reading the documentation in detail and average users may depend less upon the documentation as their familiarity with the software increases. A quick reference chart can be provided in each section of the manual to aid these users. This type of chart contains the name of a command or function, its syntax and opti ons within the syntax, and a short remark defining the command (Huntington, 1983). Many experienced computer users can perform most operations within a specific software package through the exclusive use of a quick reference chart.

Educators planning to modify software documentation should develop a checklist to serve as a guide (Sohr, 1983). This checklist should contain a minimum amount of information for inclusion in documentation manuals:

1. a detailed table of contents;
2. a complete index;
3. an introduction and summary for each chapter and section;
4. a tutorial section to provide hands-on experience to the user in a controlled environment; and,
5. a quick reference section.

Software manuals which are written or modified should contain these elements to support procedures and user-friendly guides within documentation.

**Field Testing and Implementing Documentation**

Teachers should test the documentation guide with a sample of learners who represent the intended users of the guide. Instructors should not provide coaching to users during this testing phase. Documentation should be designed to provide users with ample information necessary for successful software execution. During this testing phase, teachers should note problem areas and revise the guide to eliminate unclear structure or unclear wording. The final result should be a documentation guide which enhances the educational computing experiences of learners.

After refining the documentation guide, instructors should conduct training sessions with students to familiarize them with the guide and CBI software. When the documentation guide is put to use in the educational program, minor problems may surface which affect either the guide or the software. Instructors should establish procedures for periodically updating documentation in order to correct its deficiencies or to solve new software problems encountered by learners.

### The Future Looks Brighter

Educators can look to the future with a degree of certainty that commercially prepared software documentation is destined to improve. Most software vendors realize that their products will not maintain a market share if they don't live up to the labeling "user friendly." Additionally, these companies have the potential to utilize dramatic technical advances for creating multi-faceted documentation. Voice synthesis and recognition, interactive training with laser video discs, and infrared touch screens are technical advances that can make the documentation process an enlightening learning experience (Fawcette, 1983). However, for the time being instructors will need to think about retreading existing documentation. This process can prove to be time consuming, but it can be used to create software documentation that meets specific learner needs in a CBI program. Quality software should not be thrown in the scrap heap because of inadequate documentation.

## SUMMARY

Instructors may need to modify software documentation to correct deficiencies with vendor supplied documentation or to create documentation for custom produced software. When developing documentation, teachers should use short sentences containing a vocabulary familiar to the intended audience. Instructors should develop concrete concepts that will aid students in software execution. The guide should be organized by tasks rather than by computer commands. This will provide a "big picture" overview to the students as well as teach common commands through repetition. A detailed index and listing of error codes in simple language should be provided to help students overcome problem areas in the software. A quick reference chart should be incorporated to serve more advanced users who do not need detailed explanations of procedures. Instructors should test, correct, and retest the guide until it can function on the merit of its content. A final recommendation for documentation is for teachers to periodically update the guide to correct minor deficiencies.

## ACHIEVEMENT INDICATORS

1) Explain the term "on-line documentation."

2) Describe two types of software documentation.

3) State four common problems associated with software documentation.

4) List four writing principles that should be used to improve documentation.

5) Discuss the importance of face validity for software documentation.

6) Develop a checklist of information to be included in modified documentation.

7) Describe the process of field testing modified documentation.

## REFERENCES

Baldridge, A. (1983, November). Programming 10; documentation 3: How to improve the score. *inCider*, pp. 132-138.

Fawcette, J. (1983, December). Finally, some help for documentation. *Personal Computing*, pp. 141-143; 145.

Grimm, S. J. (1982). *How to write computer manuals for users*. Belmont, CA: Lifetime Learning Publications.

Huntington, F. (1983, April). Writing for readers. *inCider*, pp. 124-125.

Johnson, D. (1983, September). Don't forget the documentation! *Electronic Education*, pp. 66-67.

Lobello, T. (1982, November). If that's the design, chum, why isn't it in the documentation? *Electronic Education* p. 44.

Sohr, D. (1983, May). Better software manuals: Your company's survival may depend upon them. *Byte*, pp. 286; 288; 290; 292-294.

Weal, E. (1983, July). Does the manual measure up? *Popular Computing*, pp. 166-167.

Weiss, E. H. (1984, April). A messy system is still a messy system. *Data Training*, p. 23.

Wiio, O. A. (1982, June). I don't understand a word of it. *Creative Computing*, pp. 116; 118; 120; 122.

## Unit 9

# Sink or Swim: Orienting Students to Computer-Based Instruction

### UNIT OBJECTIVE

Upon completion of this unit, the learner will be able to apply concepts for orienting students to Computer-Based Instruction. This knowledge will be demonstrated by completion of the achievement indicators at the conclusion of this unit.

### SPECIFIC OBJECTIVES

Upon completion of this unit, the learner will:

1) Discuss how student characteristics and negative preconceptions can influence their initial attitudes and successes with microcomputers.
2) Describe three characteristics of microcomputers that can facilitate learning.
3) Explain the benefits of Computer-Based Instruction for students.
4) State a working definition of the term "program user."
5) Describe six steps for orienting students as program users.
6) Explain four supportive activities for orienting students to Computer-Based Instruction.

# Sink or Swim: Orienting Students to Computer-Based Instruction

BY: DR. RICHARD A. McEWING
DR. GENE L. ROTH

Remember your first attempts at learning to swim? My memory recalls a fairly unpleasant learning experience. My swimming lessons were conducted on a piano stool in the basement, under the direction of my father. Dad hoisted me onto the piano stool and proceeded to teach me all of the movements associated with swimming. I waved my arms, kicked my feet, and did all of the things I was told that swimmers generally do to stay afloat. Through Dad's guidance I became quite adept at swimming on a piano stool. When the day came to test my swimming prowess under real conditions, my confidence began to wane as I waded into the water. The water creeping around my waist caused a release of very different feelings compared to my piano stool. As a final test, my father picked me up and threw me into the water; my preceptions of water have never been the same. As I came close to drowning, I learned that piano stool swimming was not a very good orientation to water swimming.

This anecdote parallels the situation many teachers experience as they begin to integrate educational computing into curricula and instruction. Orienting students to the realities of Computer-Based Instruction (CBI) is an important first step for both teachers and students. Early perceptions can produce lasting effects. This unit will present suggestions for easing student transition from traditional learning activities to the less familiar realm of educational computing. Whether students sink or swim with CBI may depend on how well the orienting experience prepares them for educational computing.

### FIRST ENCOUNTERS WITH MICROCOMPUTERS

Some students have misconceptions regarding microcomputers. Sources of these misconceptions include parents, peers, and previous personal experiences. A common preconception of students is to overestimate the skill levels of math and science needed to effectively understand and operate microcomputers (Hannafin & Cole, 1983). This belief concerning hard science prerequisites is unfounded for most students. A more logical choice for a prerequisite to CBI is keyboarding skill (Kisner, 1984).

---

Reprinted with permission, Copyright 1984, AEDS (Association for Educational Data Systems) Monitor, 23 (3,4), 15-17; 19

Instructors may also encounter students who possess a generalized fear of microcomputers. This deeper level learning block is more difficult to attack as it can be equally frustrating for students and teachers. Such fears normally abate within four microcomputer sessions as learners begin to gain confidence that they can control microcomputer sessions (Loftus, 1982). Instructor attitudes play a significant role in helping students to reduce initial anxieties toward microcomputers. The instructor serves as a reassuring influence in the learning process. This role is congruent with accepted professional approaches used to relieve anxiety in individuals who exhibit fears or phobias. Individuals fear the unknown, especially if peers and associates lead them to suspect their abilities to succeed in new endeavors. Students' fears of educational computing can be reduced through a few positive encounters under empathetic instructors who have been sensitized to the problem.

## LEARNING STYLES AND COMPUTER-BASED INSTRUCTION

Personality and learning style affect the way people learn. Students' cognitive styles may require teachers to vary their orientation methods for CBI. Educational computing may come more easily and attract students who are considered to be introverts (Hopmeier, 1981). Additionally, field dependency may affect a student's success with computing (Stevens, 1983).

The concept of field dependency is based on the premise that individuals approach the world from one of two cognitive perspectives: (a) from a field dependent approach in which a person learns from a broad perspective and arrives at answers holistically, or (b) from a field independent approach in which learning is a step by step process and answers are logically derived. Computer technology would appear most compatible to the field independent student.

It would seem, then, that teachers should focus special attention on students who are perceived to be extroverted and/or field dependent during initial stages of CBI orientation. Students who exhibit these characteristics are possible candidates for early frustrations and, perhaps, are likely to resist CBI

## MICROCOMPUTERS AS LEARNING FACILITATORS

Seymour Papert (the creator of LOGO) claims that a scandalous outcome of education is that by **teaching** children things we deprive them of opportunities **to learn** (1984). This implies that learning involves active engagement on the part of the learner. Often, teachers control learning situations and prevent students from reaching optimum involvement. To learn, in the most real sense, students must experience control. Computer-Based Instruction, properly conceived and implemented, facilitates this type of learning. Educational computing can prevent the outcome feared by Papert if the orientation activities stress student control of the process.

Teachers should begin orientation activities by communicating to students that learning experiences with microcomputers can be positive emotional encounters. These successful encounters with microcomputers can be attributed to three characteristics which make these tools excellent facilitators of learning: (a) Microcomputers are nonjudgmental. They neither take offense at errors nor publicly ridicule students who make computing mistakes. (b) Microcomputers are consistent. Microcomputers react in consistent patterns to all people regardless of race, sex, ethnic background, religion, personality, age, or handicapping condition. (c) Microcomputers are patient. They search for the right answer, provide hints, and repeat information. They provide appropriate data to users as often as it is requested.

These three attributes of microcomputers parallel the suggested characteristics therapists might embody if they expect to enhance learning in their clients (Rogers, 1961). The preceding point neither advocated educational computing as a therapeutic tool nor does it imply that computers can replace the empathetic role of human beings. Rather, it suggests a theoretical base for the support of microcomputers as tools of learning facilitation. This belief (of the microcomputer as a learning facilitator) should be the first concept introduced to computing students. This concept should help to negate anxieties that students might harbor toward their initial experiences with CBI.

Teachers are cautioned against commenting, "Don't worry, it will be easy; everyone can operate microcomputers." The intent of the teacher may be positive, but the effect may be negative. Initial attempts to operate microcomputers may not be easy and this type of preface to instruction may serve to discourage students when mistakes occur. Teachers must be cognizant of conveying hidden messages during student orientations to CBI. These messages may be well intended; but can be counterproductive to combatting student anxieties toward microcomputers (Carey, 1984). The hidden message of the preceding is "If you can't do this, you're a real dummy."

## BENEFITS FOR STUDENTS IN CBI

The next powerful orientation step is for students to be advised of the long-term benefits to be gained by educational computing. Notice that the orientation process inaugurates CBI with a consideration of what students bring to the experience and moves ahead to consideration of what they can take from it.

The most obvious benefit to be suggested is that CBI was selected for the class because it was expected that student learning would be enhanced. Teachers should share these expectations with students. Instructors should briefly indicate why CBI was considered and adopted for curricular and instructional purposes. Thus, the first long-term advantage of CBI is that students can expect to experience a greater mastery of course content.

A second benefit for students is that microcomputer applications learned in class are likely to parallel situations students will encounter in the world of work. For example, many students will eventually work at jobs in which they will use computers for information management. A conservative estimate is that by 1990 seventy-five percent of the nation's jobs will be directly related to information processing (Montana Task Force on Computer Education, 1983). Students who experience CBI will gain an understanding of how computers function as storage and retrieval units. This general understanding will be important in the operation of office computers and should help students develop skills and concepts that will aid them in their job searches.

A third benefit pertains to increasing student skills with computer languages. Although there are numerous computer languages, they share undergirding commonalities that allow users to accomplish tasks with computers. In some cases, educational computing can provide students with opportunities to learn the exact languages they will encounter in future work situations.

A final benefit of CBI for students is the array of simulations that can expand the classroom. Simulations permit students to experience decision making. Quality computing simulations provide students with situations in which they learn how alternative decisions create favorable or unfavorable consequences. Opportunities to rerun simulation programs permit students to analyze numerous cause and effect considerations. This ability of simulation software to reflect decision-making situations in the work world may help students gain confidence in their abilities prior to acquiring on-the-job experience. Furthermore, simulations permit students to make mistakes during a learning process without undue embarrassment to themselves or costs to employers.

Teachers should employ these four benefits as a core to the orientation process. **Localization** of this segment of the orientation process occurs through teachers building onto these additional advantages of educational computing which are relevant to their individual programs, students, and professional needs.

## FAMILIARIZING STUDENTS WITH CBI

The orientation now moves back to the present with the microcomputer in front of the students. Teachers should refrain from providing too much information about computers during this orientation stage of CBI. Instructors will have gained extensive computer knowledge through developing a CBI course; they should not eagerly assume that students will need all the background information. Students will not want detailed computing information during the orientation stages of CBI. Premature in-depth presentations have counterproductive effects. In such lectures, learners are subjected to explanations which make little sense to them, which may not reach out to their needs, but which require them to reach out to the levels of the instructors, to adopt new vocabulary, and to make sense of processes about which they are totally unfamiliar (Loftus, 1982). The words "microcomputer overload" painfully depict computing initiations experienced by many people. Too many learners are told too much, too quickly. Orientation to educational computing may be crushed by a barrage of inappropriate technical jargon such as bits, bytes, RAMS, ROMS, modems, and chips.

The best way to begin is to familiarize students with basic concepts for operating software. Students are **program users** at this point in educational computing, not program writers. Depending upon instructional goals, students may not need to develop skills beyond this level of computer competency. A program user should be capable of turning the machine on and off, running a program, and being able to solve minor operational problems (Willis, 1981).

Teachers can help students move toward this level of computer competency by introducing them to operating microcomputer programs through the following six steps:

1. Introduce the brand of the microcomputer and the location of its manual. It should be stressed that brands of microcomputers feature varying operating procedures and a useful tool to consult when questions arise is a manufacturer's user's guide.
2. Introduce the key components of a microcomputer system—keyboard, monitor (t.v. screen), and storage device (tape recorder, disk drive). Discuss the care of these components and the software. This discussion should be non-threatening in that the objective is to encourage students to operate microcomputers properly so they can avoid technical failures. Teachers should limit concerns and fears. It would be useful to post guidelines for care of equipment in the front of the classroom.
3. Introduce procedures for turning on microcomputers and loading programs. An explanation of these procedures should be physically demonstrated by instructors and then students should be provided immediate practice.

Once students have been allowed to load computers and feel comfortable with this activity, they can then be introduced to the procedure of running a program.

4. Introduce students to procedures for running programs. Instructors should develop as many questions as possible for the initial explanation, demonstrate the activity, and circulate about the room as students practice. An example of such an interchange might be:

   **Teacher**: "What do you see on the monitor?"
   **Student**: "The question, 'What is your name?'"
   **Teacher**: "Try typing your name. What do you see?"
   **Student**: "My name appears as I type it. Why is it all capital letters?"
   (Teacher explains) "Now what? Nothing is happening. What did I do wrong?"
   **Teacher**: "You didn't do anything wrong, you need to tell the computer that you are finished with your response. To do so type (or press) 'RETURN' (or whatever the microcomputer in use requires)."

5. Introduce procedures used to terminate the program. Students should actually terminate their program to demonstrate that they have mastered this procedure.

6. Introduce procedures for turning off the microcomputer and leaving the area ready for the next user. Students should demonstrate their knowledge of these directions by performing the actual procedures.

Once these steps in microcomputer use have been presented in a class session, students should be permitted time to practice individually. Instructors should ask individuals to demonstrate their abilities to perform steps three through six. This orientation must pertain to the mechanics of operating microcomputers, not to the learning of course concepts.

What type of program should students use for introductory experiences? Teachers should avoid gaming programs which display impressive graphics. Such encounters may cause students to misinterpret the purpose of the orientation session and/or create a false impression that educational computing is comparable to playing arcade games. Conversely, instructors should avoid the extreme of demonstrating complex programs which will be featured at a later stage of the course. Complex programs will require students to have gained specific course content—a requirement which interferes with the goals of the beginning stages of educational computing.

Instructors should choose a simple introductory program written at the reading level of the students which uses procedures similar to what students must master for future educational computing activities. For example, if there will be simulations done on the microcomputer, then students should work through a simple simulation; if information will be presented which students must read and respond to, then students should work through this type of program. An ideal program will be one that also appeals to their interests.

A good possibility for an introductory program is one which instructs students regarding how to operate the particular brand of microcomputer they will be using. Many vendors have created these types of programs (usually referred to as tutorials) and certain manufacturers supply such programs with the purchase of their hardware. It is paramount for instructors to examine programs of this type to ensure that computing experiences are congruent with the educational goals of the course. The complexity of the program may extend beyond the skill levels of beginning students and thus be counterproductive to orientation exercises.

## PROVIDING SUPPORT FOR COMPUTING STUDENTS

To support the introduction of CBI in classrooms, a number of activities can be undertaken to enhance students perceptions of computer technology. Due to the individuality of students, instructors may find varying degrees of need for such activities. Additionally, instructors will find the time constraints and the availability of facilities will be limiting factors for the types of activities to be selected. The central concept for instructors to consider is that the selected activities should enhance the **believability** of CBI as a benefit to school work and in the working futures of the students. Suggested activities which serve this purpose include:

1. Students can practice utilizing computer printouts. Instructors can supply printouts of information which students will have interest because of the topic (e.g., sports statistics). Students can locate familiar facts in order to learn how computer printouts display information.

2. Field trips can be planned to regional businesses and industries which use microcomputers and mainframe computers. If possible, instructors should schedule sites which use the exact brands and models of computers that are available in their schools. This activity can be repeated at other times in the course to support lessons or to introduce students to other types of computers.

3. Instructors can develop games which provide practice and reinforce basic computer terminology. The nature of these games should match the maturity levels of students. Teachers can gain assistance in this type of activity by requesting students to develop games. Employing this student creation approach helps students to understand computer terms by developing games. Furthermore, this approach helps ensure that games will match maturity levels within the class.

4. Teachers can foster discussions in which students project their beliefs regarding the future of microcomputers in society. This is an excellent activity for discovering the attitudes of students toward microcomputers during their initial encounters with educational computing. Such discussions at the beginning of a course may identify students who need special attention in order to dispel misconceptions or high anxieties regarding microcomputers.

## SUMMARY

This unit provides readers with very basic, yet very important concepts regarding orienting students to Computer-Based Instruction. Early student computing experience will have a major impact on whether these students sink or swim with CBI. Students should be advised during the orientation stages of CBI that they will control their own destinies with CBI, and that microcomputers are tools for them to use for their educational benefit, both now and in the future. Once they have the confidence to float, they can progress to the various strokes of swimming.

Instructors should try to alleviate the initial anxieties students might feel toward educational computing. Students should be encouraged to begin with simple computing exercises which build confidence before proceeding to more complex CBI activities. Instructors need to be aware of how the learning styles of students can impact progress in CBI.

Some students, after initial struggles, will easily stay afloat with CBI, whereas instructors will need to watch for other students who are going under for the second or third time. Educational computing can be exciting learning for students and teachers. However, instructors need to plan orientation stages of CBI carefully to ensure that students perceive educational computing as a rewarding learning experience.

## ACHIEVEMENT INDICATORS

1) Discuss how student characteristics and negative preconceptions can influence their initial attitudes and successes with microcomputers.
2) Describe three characteristics of microcomputers that can facilitate learning.
3) Explain the benefits of Computer-Based Instruction for students.
4) State a working definition of the term "program user."
5) Describe six steps for orienting students as program users.
6) Explain four supportive activities for orienting students to Computer-Based Instruction.

## REFERENCES

Carey, D. (1984, January). Hidden messages. *SIG Bulletin*, 1 (1), 14-15.

Hannafin, M. J., & Cole, D. D. (1983). A comparison of factors affecting the elective selection of introductory computer courses. *AEDS Journal, 16* (4), 218-22.

Hopmeier. (1981, September). New study says CBI may favor introverts. *Electronic Education*, p.16-1.

Kisner, E. (1984) Keyboarding—a must in tomorrow's world. *The Computing Teacher, 11* (6), 21-22.

Loftus, W. J. (1982, September). A user's balance sheet on computer assisted learning. *Technical Horizons in Education*, pp. 118-123.

Montana task force on computer education (1983). *The Elements of Computer Education*. Helena, MT: Montana Office of Public Instruction.

Papert, S. (1984). Computer as mudpie. *Classroom Computer Learning, 4* (6), 3-40.

Richman, E. (1982). *Spotlights on Computer Literacy*. New York: Random House.

Rogers, C. (1961). *On Becoming a Person*. Boston: Houghton Company.

Stevens, D. J. (1983). Cognitive processes and success of students in instructional computer courses. *AEDS Journal, 16* (4), 228-233.

Willis, J., & Miller, M. (1981). Computers for everyone. In J. Rodenstein & R. Lanbert (Eds.) *Microcomputers: Applications in Vocational Education* (no page numbers). Wisconsin: University of Wisconsin-Madison, Vocational Studies Center.

## Unit 10

# Execute Computer-Based Instruction

### UNIT OBJECTIVE

Upon completion of this unit, the learner will be able to apply concepts for executing Computer-Based Instruction. This knowledge will be demonstrated by completing the achievement indicators at the conclusion of the unit.

### SPECIFIC OBJECTIVES

Upon completion of this unit, the learner will:

1) Discuss how Computer-Based Instruction affects classroom communication and thought patterns.

2) Develop a list of rules for students pertinent to microcomputer usage.

3) Discuss considerations for the physical locations of microcomputers in classrooms.

4) Generate ways to maintain student motivation in Computer-Based Instruction.

5) Create a classroom specific **User's Guide** for a Computer-Based Instruction program.

# Executing Computer-Based Instruction

### BY: DR. RICHARD A. McEWING
### AND DR. GENE ROTH

Historically, vocational teachers have faced the problem of keeping pace with the advancing technology of the work place. Preparing students for a changing world of work is a constant reminder to vocational teachers that they do not have the luxury of resting on previously learned work skills and knowledge. Vocational educators must keep abreast of contemporary developments within their vocational areas of expertise in order to guide students toward quality performances.

In addition to this concern for preparing students for a fluctuating world of work, vocational educators must contend with the application of new instructional technologies. Schools could be facing the most significant instructional technology challenge in the modern history of education—Computer-Based Instruction. The rapid influx of microcomputers into vocational classrooms has caught educators unprepared to effectively utilize this contemporary instructional technology. As school systems continue to acquire computer technology, numerous vocational instructors are talking, or at least thinking, about where to start with these machines (Roth & Tesolowski, in press). This unit provides vocational teachers with several key points to consider for integrating CBI into vocational education programs.

---

Reprinted with permission, *Media and Methods, 21* (7), Copyright 1985.

## Reaching a Harmonic Chord Among CBI Students and Instruments

Computer-Based Instruction drastically affects communication patterns that occur in vocational classrooms. To gain success in vocational teaching, instructors must harmonize communication patterns with their instructional practices. Vocational teachers need to examine communication patterns among students, programs, and themselves to successfully integrate CBI with vocational curricula and instruction.

## Communications of the First Kind: Student and Computer Program Interactions

CBI creates diverse communication patterns that are new to vocational learning environments. Consider the communication exchange that most likely will occur—the interaction of students and software. Students will expend considerable time and effort using microcomputers to communicate with software. What are the characteristics of these interactions? These exchanges are analogous to a conversation with a person who listens intently, never misses a point, but is inept with the subtleties of language. As a result, the person frames few suppositions about what is "meant" as opposed to what is "said."

Vocational instructors can expect comparable communication dilemmas from students who periodically complain that the computer is not performing according to the students' expectations. These types of malfunctions (student fe/male functions) are frequent initial barriers for neophytes in educational computing. Common student errors consist of misusing computer commands, misspelling words, or making keyboarding mistakes. Vocational instructors can ease the pain of students who suffer from computer fe/male functions by properly orienting students to educational computing. Vocational students must discern that microcomputers, like mirrors, reflect user errors. Vocational students can be encouraged to develop analytical habits for exposing their computing errors as opposed to blaming mistakes on computers. Instructors should stress student attitudes which are less critical toward the eccentricities of computers and more demanding of themselves.

Vocational students who enjoy Computer-Based Instruction will strive to understand the underlying logical progressions of educational computing. Whereas humans are capable of formulating mental leaps to reach a desired solution, computers must track a problem solution by pursuing step-by-step process. Some students may identify with this required step-by-step process by recalling situations in math classes in which they were unable to solve problems. An understanding of these problems occurred when the teacher identified small steps that had originally been omitted because they were considered obvious (Howerton, 1982).

## Communications of the Second Kind: Student to Student Interactions

An initial impression might be that educational computing in vocational classrooms decreases interaction among students. Each student is envisioned intently working at a microcomputer station. This vision is an unlikely reality and is not desirable from a number of points of view.

Purchases of microcomputers by school districts have been increasing at an approximate rate of fifty percent per year. Future Computing, Inc., a Texas based research firm, has predicted that the combined market for educational hardware and software will reach one billion dollars per year by 1987 (Watt, 1983). However, these numbers are relatively miniscule compared to the total numbers of students and classrooms to be served. If this fifty percent increase in microcomputers is sustained to 1986, there will still only be approximately three microcomputers per school or one for every eight classrooms (Becker, 1982).

For most vocational programs it is fiscally impractical for each student to have unlimited access to a microcomputer. However, unlimited access to a microcomputer is not required for successful educational computing. Research findings suggest that as few as ten minutes of computer use per day result in increased levels of motivation, rates of learning, and lengths of retention. Additionally, findings indicate that up to four students working at one machine gain comparable educational benefits to one per computer (Fisher, 1982). These research findings suggest that an educational computing approach which features students working at computer stations in team structures is a viable process for CBI in vocational programs. Team concepts will require vocational teachers to plan for various types of student interactions that can enhance CBI.

In examining the interdynamics of computer teams, several questions should be addressed by instructors: (a) Should teams be composed of students of similar backgrounds or should each team have at least one experienced user? (b) Should teams be grouped on the basis of career interests? (c) Should teams be formed according to academic abilities?

There are limited benchmarks to help vocational teachers answer these and similar questions. However, such problems can be minimized if team formation is not viewed as a one time occurrence which locks students into patterns for the entire school year. Curricular content and instructional methods may prompt the formation of various types of team compositions. Vocational instructors are encouraged to experiment with a number of team structures to keep students from becoming either too dependent or too dominant in their groups. Communication should be allowed within teams and between teams to develop a healthy competitive spirit. Roles of instructors transform, to that of learning facilitators as teams work independently and collectively to perform educational computing tasks.

### Communications of the Third Kind: Student and Teacher Interactions

Monitoring student to student interactions in CBI will be challenging for vocational instructors, but not more so than teacher and student interactions. Two communication pitfalls await vocational teachers in the implementation of CBI: (a) instructors may inadvertently rely on traditional teaching methods that had been employed prior to the advent of CBI, and (b) instructors may remove themselves from interactions with students by perceiving that software will manage instruction.

The first pitfall could occur if vocational teachers fail to recognize that students, through their use of microcomputers to interact with coursework, will have an increase in individual needs. Although vocational students may be pursuing common educational goals, their abilities and rates of comprehension vary. The teaching focus of CBI should be on the team or individual, not the entire class. Once the organizing directions are given to students, it will be unproductive to make whole class presentations on material unless it can be shown to be relevant to the entire group. Vocational teachers will be able to facilitate learning by circulating among microcomputer stations, monitoring student interactions, encouraging progress, checking on hardware, trouble-shooting problems, and reviewing software usage.

Vocational students benefit from such personal interactions that keep them motivated and headed in the correct direction. Often students will need to discuss information and procedures in more detail than what is presented in the hardware and software documentation. It is through these individual conversations at microcomputer stations that learning will be enhanced. Effective interactions between instructors and students will be the keystone for successfully orchestrating CBI. Optimal student and teacher interactions in CBI will create the harmonic balance between "high tech/high touch" (Naisbett, 1983).

## STUDENT RULES FOR COMPUTER-BASED INSTRUCTION

Several anecdotes have been used to convey the disapproval of rules. The continuum of anecdotes begins with an unspoken belief—"rules are made to be broken"—and ends with the golden rule—"whoever has the most gold"—rules. To avoid nurturing negative perceptions of rules, vocational teachers should not develop and present CBI rules to students as "necessary evils." Rules need to be constructed, but their existence should provide a positive contribution to class expectations. Clear rules, accompanied by a clear explanation of the rules, will enable students to avoid situations which could be damaging to themselves or equipment.

Rules for guiding student actions in educational computing may be grouped into three clusters: **access, safety,** and **ethics.** The first cluster, access rules, requires students to possess procedural skills and knowledge prior to operating computers. Access rules need to be established concerning (a) which students at what times will be permitted to work at microcomputer stations, (b) food and drink restrictions, and (c) game playing policies. These procedures should be written, distributed, explained, and posted in the classroom. Vocational teachers should solicit class discussion on each item. Student input should be valued by instructors and utilized in revising CBI access rules, as class experiences will foster rule revision and modification.

The safety rule cluster should be non-debatable. Safety rules include information related to power cord and plug-in locations, movement of equipment within the room, control of static electricity, alteration of hardware connections, reduction of glare and dust, and maintenance of an appropriate temperature for the microcomputer environment. Safety rules, like access rules, should be posted and clearly visible to vocational students at their microcomputer stations. The uniqueness of each vocational education laboratory or classroom may necessitate additional discipline related regulations.

The final cluster of rules pertains to the ethical application of microcomputers. Vocational teachers should not find themselves accused of giving students skills without giving them insight into ethical concerns regarding the use of those skills. For example, software pirating is a major problem for manufacturers (Harrison, 1983). Vocational students will quickly learn processes for duplicating programs for their personal use. Such duplication of software is a form of stealing if the program is not in the public domain. Vocational teachers must do more than talk about computer ethics. Instructional materials that support and supplement CBI must demonstrate an acceptance of computing ethics. No computer materials should be permitted in the classroom if they have been obtained by unauthorized procedures. Additionally, instructors should not permit the reproduction of copyrighted software by students or faculty using vocational education computer systems.

Enforcement of these three clusters of rules will require well defined consequences for rule violators. The most relevant consequence would be denial of access to school computers. Educational computing can be fun for teachers and students, but guidelines must be enacted to ensure that instructional activities are meaningful and purposeful for students.

## SETTING THE STAGE: PLACEMENT OF MICROCOMPUTER WORKSTATIONS

Several considerations for locating microcomputer workstations pertain to environmental factors that may affect the performance of microcomputers and their users. Typically, microcomputers function best in locations that have good temperature control. Microprocessors transmit heat. As a result, a small room crowded with students and microcomputers could create a hot, stuffy, unfavorable learning environment.

Conditions that need to be controlled include static electricity, dust, and glare. These elements, unlike the problem of temperature control, should not require extensive modification of existing facilities. Many manufacturers offer recommendations for controlling these problems. For example, some manufacturers suggest that chairs without rollers be used in microcomputer laboratories to decrease levels of static electricity.

An equally important concern when planning the integration of CBI into vocational programs is achieving a balance between computer accessibility and security. While the placement of computer hardware to limit the potential for vandalism and misuse is necessary, there is a danger of protecting equipment to the extent that it is inaccessible to students and faculty.

Vocational instructors should direct their attention to several key questions regarding the locations of microcomputer workstations: (a) Will students use microcomputers only during class time? (b) Will students be expected to perform computer related assignments at times other than class periods? (c) Will students other than vocational students have access to microcomputers? (d) Will students be working in teams (requiring greater space allocation per computing station)? (e) Will hardware be bound to workstations?

Answers to the preceding questions will have continuing impact on the execution of CBI. Vocational instructors should avoid unnecessary precautions that tend to turn students away from microcomputers. Teachers should strive to achieve maximum flexibility for computer usage with minimal risk to facilities and equipment. The overriding intent is to maximize potential uses while minimizing potential abuses.

## BOLSTERING STUDENT MOTIVATION

Aware of the inseparable connection between motivation and learning, vocational instructors rightly pursue relevant, challenging, and exciting vocational curricula and instruction programs. Several writers (Chandler, 1983; Jernstedt, 1983; Peelle, 1982; Stowbridge & Kugel, 1983) have portrayed CBI as a method for motivating learners. Nevertheless, CBI will not motivate students without sound educational objectives. Enhancement of motivational levels will be achieved through approaches employed by vocational instructors to orient students to CBI objectives.

The orientation process should feature a cycle of pre- and post-computer activities for building student motivation. For example, prior to each unit of instruction vocational teachers should set the stage for the educational objectives of the unit. This procedure is frequently called **set induction** (Kim & Kellough, 1983). Students should be introduced to a unit of instruction in a manner that stresses the importance and significance of the curriculum content. Students become aware of what will be learned in what fashion, and individually they become stimulated to explore the unit. After set induction, students begin to work while the instructor circulates and interacts with students in a facilitating role.

One motivating educational system, a method ideally suited for integrating CBI into vocational education curriculum and instruction, is competency-based education. This systematic approach to curriculum and instruction consists of three basic steps (Sullivan & Higgins, 1983). The first step involves the teacher establishing meaningful and worthwhile student objectives. During step two, the teacher develops appropriate instructional activities to mesh with learner objectives. In step three, the instructor monitors the progress of students toward achieving the educational objectives. This competency-based approach permits instructors to allocate more time and effort for working on an individual basis with students.

During this execution phase, it is useful to require students (as individuals or teams) to maintain and submit daily or weekly logs. This computer diary relates student experiences with specific computer software on given dates and times. From this information, vocational instructors may garner a sense of problematic software as well as rates of progression for individuals and teams. Information gained from student log books will provide evaluative feedback to instructors for modifying CBI.

Debriefing sessions should be used to conclude units of instruction. Students should receive feedback regarding how well they executed the unit. Sessions should include a sharing of information among students and teams. Additionally, debriefing could include reviewing microcomputer skills or information outlined in previous instructional units. This cycle of briefing, executing, and debriefing ensures that goals of CBI are achieved and helps students recognize their progress and direction.

In addition to methods which focus on building intrinsic motivation of students, vocational instructors need to examine methods for extrinsic motivation. Extrinsic motivation should incorporate activities within the structure of the classroom that add fun to vocational education and create positive effects on work outputs.

Boss (1983) describes an innovative adaptation of CBI that increased the motivational levels of her students. She created the "Order of the Apple," which aroused student interest by fostering friendly classroom competition to attain six orders. Upon completing each level, students receive a different membership card. An interesting sidelight to this competition is the **bump command,** which permits students who are solving specific, teacher-directed problems to bump students off computers if they are merely playing games.

## CREATING A USER'S GUIDE

A pivotal consideration for vocational instructors is the creation of a microcomputer user's guide. Initially, this idea might appear worthy of little consideration because manufacturers provide user's guides. Frequently, and unfortunately, the expression "user's guide" is a misnomer for such documentation. Computer documentation provides a disproportionate amount of information that can easily be misinterpreted by a person of limited computer vocabulary, i.e., the CBI novice. Readability tests for most hardware and software documentation would reveal a comprehension level beyond that of many students. It is therefore recommended that vocational instructors extrapolate information from documentation manuals and construct a user's guide that fulfills the computing needs of their vocational students.

A program specific user's guide should not be inclusive, but it should contain four components. In the first section, teachers should include diagrams of the microcomputer system and its components. Technical terms and brand names should be included. The second section should contain directions pertaining to basic computing skills such as loading, unloading, and turning on and off power sources. Third, an overview section of appropriate computer terminology should be provided to help students expand their computing vocabulary. Fourth, the guide should contain a trouble-shooting section to help students correct simple malfunctions and react to program error messages.

An example of a school/program specific user's guide is the *Apple II Do-It-Yourself Manual*, written by the Hopkins public schools of Hopkins, Minnesota. This type of manual can serve time-tested CBI needs for vocational students and instructors. The user's guide should be consumable and specific to the classroom or laboratory. Manufacturers' user guides will remain useful to students who seek additional familiarity with computing.

## ENHANCING THE EXECUTION OF CBI

Vocational instructors can ease the transition of CBI into vocational curricula and instruction by employing educational computing activities which extend vocational concepts that are components of the program. Integration of computing into the vocational club may increase interest in CBI as well as vocational club activities. Students could be requested to help develop CBI materials for the management of club activities or for incorporation into the vocational program.

Another activity which will enhance CBI in vocational programs is discussion regarding computer ethics. One method for fostering such discussions is to create a computer news bulletin board for posting newspaper clippings and magazine articles related to computer controversies. Such controversies should include issues of software piracy, alteration of computer records, copyright restrictions, ramifications of computer technology to the world of work, and the impact of high technology jobs on personal life styles. Students should be encouraged to supply articles for discussion purposes. As controversial news items are considered, different formats for presentation should be used in vocational classes. Debate, role playing, guest presentations, media presentations, and group decision making are techniques that maximize student interactions on controversial computer topics.

One particularly effective method of exploring computer ethics is to present students with short dilemma stories about fictitious characters. A dilemma story depicts a situation in which the characters have at least two attractive courses of action. Through discussions of alternative solutions to these dilemmas, vocational students can clarify their values while developing reasoning and decision-making skills. Not only will students enjoy the computer ethics activities, but vocational instructors will gain insight regarding students' thought processes and value systems (Hannah & Matus, 1984). The central idea of all such enhancing activities is to help students reach an awareness of the complexity of computer ethics. Through such activities students are encouraged to examine alternative courses of action and to consider the consequences of those actions.

## SUMMARY

Amid fanfare and high expectations from school administrators, school board members, parents, teachers, and students, Computer-Based Instruction is being implemented in vocational programs across the nation. Vocational teachers will play pivotal roles in the success of these programs. Teachers will carry the brunt of responsibilities for integrating CBI into vocational curricula and instruction. Instructors will face the challenges of synthesizing student talents with instructional materials, hardware, and software. Teacher/student/computer interactions, the educational computing environment, and the instructor's own perspective of educational computing must be considered during the implementation stages of Computer-Based Instruction.

## ACHIEVEMENT INDICATORS

1) Discuss how Computer-Based Instruction affects the three classroom communication and thought patterns.

2) Develop a list of rules for students pertinent to microcomputer usage.

3) Discuss two considerations for the physical locations of microcomputers in classrooms.

4) Generate three ways to maintain student motivation in Computer-Based Instruction.

5) Create a classroom specific USER'S GUIDE for a Computer-Based Instruction program.

## REFERENCES

*Apple II Do It Yourself Manual* (no date). Hopkins, MN: Hopkins Public Schools.

Becker, H. J. (1982). Roles for microcomputers in the 1980s. *NASSP Bulletin*, 47-52.

Boss, J. A. (1983, October). Order of the apple. *The Computing Teacher*, pp. 41-43.

Chandler, P. (1983, November). How to keep your kid's nose to the grindstone. *Personal Computing*, pp. 110-111; 113; 213.

Fisher, G. (1982). Computer use in self-contained classrooms. In J. Rodenstein & R. Lanbert (Eds.) *Microcomputers in vocational education* (pp. 319-320). Madison: University of Wisconsin, Vocational Studies Center.

Hannah, L., & Matus, C. (1984, April/May). Teaching ethics in the computer classroom. *Classroom Computer Learning*, pp. 32-33; 35-36.

Harrison, P. L. (1983, June). Software piracy. *Computers 83*, pp. 41-44.

Howerton, D. A. (1982). Integrating microcomputers into marketing and distributive education. In J. Rodenstein & R. Lanbert (Eds.) *Microcomputers in vocational education* (pp. 275-278). Madison: University of Wisconsin, Vocational Studies Center.

Jernstedt, G. C. (1983). Computer enhanced collaborative learning: A new technology for education. *T.H.E. Journal*, *10* (7), 96-101.

Kim, E. C., & Kellough, R. D. (1983). *A resource guide for secondary school teaching*. New York: Macmillan.

Naisbitt, J. (1984). *Megatrends*. New York: Warner Books.

Peelle, H. A. (1982, September-October). Computing vitamins. *Educational Computer Magazine*, pp. 24-25; 76.

Roth, G. L., & Tesolowski, D. G. (in press). The microcomputer freeway: Mapping a future for vocational curricula. *The Journal of Epsilon Pi Tau*.

Stowbridge, M. D., & Kugel, P. (1983, April). Learning to learn by learning to play. *Creative Computing*, pp. 180; 183-184; 186; 188.

Sullivan, H., & Higgins, N. (1983). *Teaching for competency*. New York: Columbia University, Teachers College.

Watt, D. (1983, August). Computers in education. *Popular Computing*, pp. 82-84.

## Unit 11

# Individualizing Instruction With Computer-Based Instruction

## UNIT OBJECTIVE

Upon completion of this unit, the learner will be able to individualize instruction with Computer-Based Instruction. This knowledge will be demonstrated through completion of the unit achievement indicators.

## SPECIFIC OBJECTIVES

Upon completion of this unit, the learner will be able to:

1) Explain how the learning concepts of attention and competence influence motivation and hence any program of individualized instruction.
2) Describe characteristics common to the tutor and the microcomputer program as they relate to individualized instructional goals and practice.
3) Define the term **user friendly software** and identify packages which embody this characteristic.
4) Distinguish reinforcing microcomputer programming practices from those which are not reinforcing to individualized instruction.
5) Create an individualized microcomputer program within a classroom setting with consideration given to scheduling, computer aides, and control.

# Individualizing Instruction With Computer-Based Instruction

### BY: DR. RICHARD A. McEWING
### DR. GENE L. ROTH

## THE BASIC NEEDS FOR ATTENTION AND COMPETENCE

In discussing ways that vocational teachers might structure classes to provide individualized instruction, it is worthwhile to consider a perspective regarding how students are motivated to learn. If the teacher keeps in mind some underlying idea pertinent to what motivates students to learn, then this guiding principle can suggest numerous ways to achieve the classroom environment necessary for learning to occur.

Studies by Page (1958) and Thistlethwaite (1959) suggest that students are positively influenced by attention. Page's work indicates that when students were provided written feedback on their papers, their subsequent performance improved. Thistlethwaite's study of high school graduates found that those who had received the most recognition for a similar achievement had the most favorable attitude toward intellectual activities.

These two studies confirm what most educators believe . . . if students are expected to improve their knowledge—if they are expected to learn—then they should be encouraged through the use of external reinforcement. Furthermore, the more extensive the positive feedback, the more positive feelings the students are likely to exhibit.

Is positive attention enough to motivate learning? Some theorists suggest not. While this external condition of positive attention is important, an internal condition is of even greater importance—the need for competence.

For one to be really involved in the learning process, the activity must be intrinsically motivating. It is suggested that "mastery motive" comes to influence learning behavior. It is the strongest and most dependable motivation for learning. This feeling of being able to deal with an important new area of knowledge and of self

---

Reprinted with permission, copyright 1985, *Educational Technology*.

satisfaction enables the student to manipulate the learning environment effectively and cope with problems which arise in the learning process. This perspective of learning development is based on the work of Jerome S. Bruner (1961).

Using these concepts of positive attention and competence as a base for motivating learning, the vocational teacher needs to select an instructional strategy which maximizes these two events for the student. One such strategy is individualized instruction.

## INDIVIDUALIZING AND TUTORING

Attention giving and competence development appear to be maximized in situations where there is one teacher and one pupil. This conceptualization of individualized instruction is not new. Many affluent members of societies prefer that their children not be educated in common classrooms. Rather, they prefer that a tutor be hired to give their children "individual" attention. These parents expect their children to benefit by (a) having a carefully selected authority personally available (many tutors are skilled in an area—military, letters, statesmanship), (b) having this availability immediate (tutors often live on the estate), and (c) having immediate reward and punishment administered according to learning outcomes (many tutors were quite cruel to their charge, with the consent of parents).

Other citizens have extolled the values of the common public school. In these schools, it is argued, students learn the intricacies of society. They learn to respect the contributions of others in building society, they learn to be patriotic, and they learn a wider range of skills. Common schools are not limited to the knowledge of one teacher; thus, students have a larger world at their disposal. The drawback of the classroom as a place of instruction is that the individual may become lost in a lock-step approach among the rest of the students. The student may go for days, perhaps even weeks or months, without responding to the instructor's questions. Many students avoid asking questions that might draw attention to them, fearing a loss of self-esteem. The strategy of individualizing instruction is intended to deal with this problem associated with whole class instruction.

The idea at the center of individualized instruction is to return the focus of instruction to the individual. Rather than one teaching strategy, it is a group of strategies aimed at improving the individual's interaction, in terms of quality and quantity, with the subject matter. Typical ways to achieve this goal involve:

1. structuring time flow in the classroom,
2. structuring subject material, and
3. structuring teacher expectations.

## COMPUTER AS MODERN DAY TUTOR

### The Presentation Element

One of the biggest complaints voiced by students is that the teacher is unavailable when needed. Just when the student needs a problem to be clarified, the teacher is over in another part of the room. The student has no recourse except to wait. Jackson (1968) made observations in elementary school classrooms which suggest that waiting is the most common activity students perform at any given time.

Use of the microcomputer can alleviate this problem. A good microcomputer program has built within it a useful series of prompts and help statements. Vocational teachers snould look for these two characteristics when purchasing classroom programs. Nevertheless, if the classroom terminal is connected to a mainframe computer, the students may experience some slow response time or some down time. These "attention lapses" are bothersome, but minimal compared to the attention lapses in a regular classroom. In addition, the students are not likely to take these down times personally. After all, it would be hard to believe that a machine is intentionally failing to acknowledge the requests of students because they are not liked. However, lapses from teachers are not so rationalized. Most computer centers plan their maintenance down time so that the user can work around these periods. Thus, like the tutor, the microcomputer offers increased availability of instruction to students.

"Prompts" and "Helps" relate to a second function that an individual tutor can perform. Not only is the student being given attention in terms of time commitment, but this attention can be programmed to be very positive. Computer programs which seem to take into consideration the feelings of a novice operator are termed **user friendly.** This is another important characteristic that instructors should seek when selecting a program. If the student is given a question and selects the wrong response, what should a good program do? The more helpful and interactive the program, the better. The following is an example of such a user friendly program in action:

Scene: Student is asked to read some text materials on the screen and then answer some questions.
Program: "Would you like to read the material again?" (yes/no)
Student: "Yes"
Program: Shows material, displays the question again.
Student: "No"
Program: Asks first question.
Student: Inputs answer (answer is wrong)
Program: "Sorry, your answer is incorrect. Would you like to try again?" (Yes/No)
Student: "Yes"
Program: Prints the question again.
Student: Inputs the correct answer.
Program: "Right, would you like to try another question?" (Yes/No)
Student: "Yes"
Program: Prints the second question.
Student: Inputs the wrong answer.
Program: "Sorry, the answer is incorrect. Would you like to try again?" (Yes/No)
Student: "Yes"
Program: Prints out the second question again.
Student: Inputs the wrong answer again.
Program: "Sorry, the answer is incorrect. Would you like to see the part of the text that pertains to this question?" (Yes/No)

### The Knowledge Element

The provision for positive acceptance extends beyond being user friendly. This extended concept ties directly to the concept of individualization. In using microcomputers to facilitate individualized instruction, one should look closely at how the software program provides for **reinforcement** of the task at hand. Swenson and Anderson (1982) stated that reinforcement, by definition, must be individualized, since what is reinforcing to one student may not be reinforcing to another. They suggest that the four most important variables influencing the effectiveness of the reinforcer are timing, appropriateness, relevancy, and configuration.

1. Timing—To be effective, the reinforcer should occur immediately after the desired response (.5 seconds is ideal); it should then be "thinned out" over time (occur every other time, then every fifth time, etc.) and not linger on the screen more than a few seconds.
2. Appropriateness—To be effective, reinforcers must recognize the age level and maturity of the user. The best approach is to test reinforcers on intended audiences.
3. Relevancy—To be effective, reinforcers should add to the user's information on the subject.
4. Configuration—To be effective, reinforcers should be part of the program as a whole and not just additions. They should deal with student errors in a positive way without judgmental information which is not only unnecessary, but undesirable.

Another characteristic of the tutor situation is the limited knowledge available to the student. These limitations in subject area have been aptly pointed out by the advocates of the common school with its myriad of specialists and specialties. Unlike the tutor, the microcomputer is not fixed by what is there. A computer may teach one thing to one person and, by changing programs, teach another thing to someone else. The real advantage for the student lies in the ability for learning to take place through one conduit. The student does not have to worry about not getting along with teacher so-and-so, thereby being turned off to that subject area. If the student has an interest in an area and access to a microcomputer and computer program, the knowledge is available. The use of the microcomputer combines the best of the tutor and common education. The student has but one "authority" to learn through, yet has available the knowledge of a much wider world.

Ryba and Chapman (1983) conclude that many effective learning programs have characteristics which can easily be built into computer software. These characteristics include allowing the student to direct the program at several points with options such as "advance," "quit," "repeat," or "review." Good software programs provide for student goal setting and self-evaluation (e.g., "The last time you had 10 correct answers, how many correct responses do you want to attempt this time?"). Additionally, software programs may allow users to pick from a list of personally meaningful rewards and to verbalize one's findings in teams or small groups. Software packages which are particularly attention drawing are multi-sensory in nature. Programs which use graphics, sound, and color are more likely to keep students' interest longer than bland programs. Similarly, programs which provide a variety of reinforcers are likely to be more favored by students.

### The Control Element

The last characteristic of the tutor, that of role model, would seem to be lost in the computer age. Surely, instructors do not wish and perhaps even fear, for students to identify with machines. Rather than despairing over this role model failing on the part of the computer, teachers should be encouraged. Vocational instructors need not be concerned about negative personality influences interfering with the presentation of material. A well written computer program neither ignores nor alienates students. The microcomputer does not attempt to make the student like it or approve of it, as a tutor/mentor is likely to attempt.

Some teachers believe that a human element can be returned to microcomputing through the use of more advanced students as computer aides. These students can help novices troubleshoot simple program and device problems. Special aides need to be used wisely, however, because their presence could have negative effects on the learning of the class (Glass, 1981). Student aides may create frustrations and resentments if they function from positions of power and privilege. Furthermore, aides may prevent students from gaining insights on their own, an experience which can be highly motivating. An alternate approach to increase interaction among students is to use group work situations for mutual support—discussions, cooperations, joint projects.

How do students perceive individualized instruction via microcomputers? Frizot (1980) found Computer-Based Instruction to be popular with students:

> Conceived as an individualized instruction in the time-sharing format, the student rightly feels that he is spending his own time and applying himself to a project of which he controls some of the parameters and also that he can, in numerous cases, make moves and take alleys of learning on his own, without disrupting the rest of the class. Thus, the student's work is individualized but not solitary. (p. 107)

## INDIVIDUALIZING THE CLASSROOM

The concept of individualizing control may be the key to successful individualization with Computer-Based Instruction. Vocational teachers can extend this control factor by individualizing to an extent that permits students to choose among software packages. Once students select software programs, there should be options within the software packages to accommodate learners at different levels. The sequence of presentation should allow advanced students to skip material, but the program should also provide avenues for return and review. Students should be free to choose the number of exercises of a particular type they wish to attempt.

For students who are skeptical about the ability of students to make appropriate self selections, there is a middle ground wherein one can use pretests to place students with scheduled mastery tests at specified check points. Programs can be organized into modules within a flowchart format involving specified objectives. Well written modules will allow for maximum student choice and self-pacing.

An individualized program approach will necessitate close attention to the scheduling of microcomputers. Blocks of time should be scheduled for class use and for free time use. Students should sign up for free time use in advance, but should be guaranteed regularly scheduled use. Since it is likely that more than one student will use each microcomputer, a log sheet should be maintained to indicate computer use and notations regarding problems with equipment or software. There should also be a provision regarding "wait time." If a student fails to show at the appointed free computing time, a provision should be established regarding how long another student should wait before taking over that time slot. These types of rules should be clearly posted in the microcomputer facility.

## SUMMARY

Computer-Based Instruction is, in many ways, the best facilitator of individualized instruction next to the tutor; perhaps it is even superior. User friendly software programs that provide positive attention, build feelings of competence, and allow extensive student control are best suited for individualized instruction. Educators may choose to individualize within the software programs or they may choose to extend this individualization to the structure of the classroom flow itself by providing increased flexibility in scheduling, program selection, and student control.

## ACHIEVEMENT INDICATORS

1) Explain how the learning concepts of attention and competence influence motivation and hence any program of individualized instruction.
2) Describe the characteristics common to the tutor and the microcomputer program as they relate to individualized instructional goals and practice.
3) Define the term **user friendly software** and identify packages which embody this characteristic.
4) Distinguish reinforcing microcomputer programming practices from those which are not reinforcing to individualized instruction.
5) Create an individualized microcomputer program within a classroom setting with consideration given to scheduling, computer aides, and control.

## REFERENCES

Bruner, J. S. (1960). *The process of education.* Cambridge: Harvard University Press.

Frizot, D. (1980). Future shock for today's education: Introduction of a computer in a French secondary school. In E. D. Tagg (Ed.), *Microcomputers in secondary education* (pp. 105-109). New York: Elsevier North-Holland.

Glass, S. (1981). Using computer-wise students . . . wisely. *Electronic Learning, 1,* 26-27.

Jackson, P. W. (1968). *Life in Classrooms.* New York: Rinehart & Winston.

Page, E. B. (1958). Teacher comments and student performance: A seventy-four classroom experiment in school motivation. *Journal of Educational Psychology, 49,* 173-181.

Ryba, K. A., & Chapman, J. W. (1983). Toward improving learning strategies and personal adjustment with computers. *The Computing Teacher, 11*(1), 48-52.

Swenson, R. P., & Anderson, C. A. (1982, October). The role of motivation in computer-assisted instruction. *Creative Computing,* 134-139.

Thistelthwaite, D. L. (1959). Effects of social recognition upon the educational motivation of talented youth. *Journal of Educational Psychology, 50,* 111-116.

## Unit 12

# Assess Student Microcomputer Skills

## UNIT OBJECTIVE

Upon completion of this unit, the learner will be able to develop an assessment instrument for evaluation of student microcomputer skills. This knowledge will be demonstrated through completion of the achievement indicators at the end of this unit.

## SPECIFIC OBJECTIVES

Upon completion of this unit, the learner will be able to:

1) Define computer literacy in terms of vocational education requirements.
2) List at least five general microcomputer skills which students should possess in a computer-based vocational program.
3) List four steps required in a computer familiarization course to normalize students' microcomputer skills.

# Assess Student Microcomputer Skills

BY: MICHAEL T. DROTTER

## Introduction

Not too many years ago, vocational instructors could safely assume that incoming students had no previous microcomputer skills. Today, however, a group of students entering a vocational discipline may exhibit a complete spectrum of computer skills, from no experience to ten or more years of computer use. This wide range of computing skills requires vocational educators to change strategies for educational computing to ensure that all students possess a minimum set of microcomputer skills prior to starting a computer-based vocational program. This minimum set of microcomputer skills may be referred to as computer literacy.

## Computer Literacy

Computer literacy, a highly overused buzz word in educational computing, has been given many different definitions by educational authors. Perhaps the most common definition of computer literacy is the ability of a person to use a microcomputer to perform routine tasks required in a job. This definition implies that knowledge and experience with microcomputers makes a person computer literate. Using this definition during the pre-assessment phase of determining students' microcomputer skills may create an inaccurate evaluation. Due to the non-standard nature of the computer industry, many individuals who are extremely proficient in the use of one microcomputer system may feel hopelessly illiterate when seated at another microcomputer's keyboard. The vocational educator must, therefore, redefine computer literacy for the purpose of assessing student microcomputer skills.

For the vocational educator, computer literacy is a general knowledge of microcomputers that aids the student in learning to use a particular microcomputer. This definition implies that the vocational educator must assess both the student's general knowledge of microcomputers and the student's specific knowledge of a particular microcomputer system. Inevitably, the microcomputer will be used directly in developing student mastery of minimum microcomputer skills. Manuals, textbooks, or video training series only provide the background information necessary to achieve a mastery of basic microcomputer skills. The logical result of this assessment will resemble a CBI program for basic microcomputer skills. Techniques illustrated in units C.9 (Orienting Students to CBI) and C.3 (Develop Lesson Plans Incorporating CBI) of this handbook provide detailed strategies for vocational educators which can be applied directly to normalizing student microcomputer skills prior to beginning a vocational CBI program.

## Establishing Assessment Procedures

An initial step in devising a microcomputer skills assessment instrument is to determine the objectives students must meet to prove competence in basic microcomputer skills. The assessment instrument must address both general microcomputer knowledge and specific knowledge of the particular microcomputer systems in use. This assessment will help vocational teachers determine the levels of effort required for students to achieve minimum acceptable microcomputer skills. The students' general microcomputer knowledge is a good indicator of how quickly they can assimilate specific skills necessary to operate in a particular CBI environment. Assessment of general microcomputer knowledge will fall mostly in the cognitive domain. General knowledge of specific microcomputer components, such as disk drives, CPUs, and software, can easily be adapted to use on a specific microcomputer system. Knowledge of microcomputer terminology will aid students in the use of reference material associated with specific microcomputer systems.

## Assessment of Specific Microcomputer Skills

When establishing objectives regarding minimum skills that students must possess to operate the CBI microcomputer system, care must be exercised to include only those skills required for the specific computing tasks which have been integrated into the vocational program. For example, if no computer programming will be performed by the student during CBI, then vocational instructors should not evaluate student's knowledge of programming languages. Likewise, instructors should not expect vocational students to acquire an extensive knowledge of the computer's operating system if students will be using "canned" commercial software in the vocational program. Appendix A provides a checklist of recommended microcomputer skills for the educational computing activities of a vocational program. This checklist should be used as a starting point to evaluate which skills are necessary for specific CBI applications.

The use of microcomputers involves psychomoter skills. Eye-hand coordination is an important consideration in the proper operation of a computer system. The only way for instructors to evaluate these skills is to place students in keyboard operating activities.

### Method of Assessment

Vocational instructors should keep traditional test-type assessments to a minimum and maximize hands-on evaluations of microcomputer skills. A method to help vocational teachers determine how quickly a student will adapt to a new microcomputer system is to limit the amount of teaching during the assessment period. This will permit instructors to ascertain how quickly students learn required skills and also allow instructors to identify low frustration thresholds students may exhibit toward technology equipment. These two bits of data may be valuable indicators regarding whether students will be able to use the standard familiarization training for CBI or if it will be required to provide individualized instruction. Teachers should assess students' skills with all the equipment that will be necessary for specific computing activities within a vocational course or program. For example, instructors should not evaluate skills with basic microcomputers when job-specific computer applications will require equipment such as a barcode reader in a retail sales program, a digitizer in a drafting program, or a plotter in an advertising program.

### Evaluate the Assessment Results

Evaluation of computer skills for CBI is really no different from standard methods of evaluation which have proven effective over the years. Pre-assessment, followed by a standardized or individualized teaching program, with built-in post-assessment and provisions to reteach certain skills, will prove to be an effective evaluation process. This process will assure vocational teachers that students who enter a computer-based vocational program will possess at least a minimum skill level in microcomputer use.

Whether computer familiarization training is implemented in a group or individual environment is best determined by follow-up CBI training. If the CBI program is one in which individualized instruction is utilized, the individualized computer familiarization is probably best. If a single class starting date is important to the overall CBI program, then group computer training is best. However, if group training is pursued, provisions must be made to provide constructive activities for students who meet objectives early, and also for students who possess prerequisite computer skills without familiarization training.

## SUMMARY

Ensuring that all students entering a CBI vocational education program possess a minimum amount of skills on the CBI microcomputer system is a key element in the success of a computer-based vocational program. Requiring all students to complete a familiarization course may be a waste of time for select students. However some students will have little or no experience with microcomputers prior to commencing CBI. It is imperative that teachers who integrate educational computing with vocational curricula and instruction develop a method to assess the level of microcomputer skills of students. Teachers should be prepared to provide remedial training for students who lack the minimum-required skills. The assessment instrument should provide a hands-on evaluation of students' skills as well as built-in indicators to predict the adaptability of students to CBI.

## APPENDIX A

### BASIC MICROCOMPUTER SKILLS CHECKLIST
### GENERAL KNOWLEDGE

**Definitions:**

ASCII collating sequence

Database

Database Management System (DBMS)

Dot Matrix Printer

Fixed (or hard) Disk

Floppy Disk

Keyboard

Letter Quality Printer

Mouse

Plotter

Random Access File

Record

Sequential Access File

Spreadsheet

Word Processor

## SPECIFIED MICROCOMPUTER SYSTEM KNOWLEDGE

How to initiate (boot) systems operation.

How to change the default disk drive.

How to load a program from disk.

How to save a program or data to disk.

How to make backup copies of programs or data.

How to halt program execution.

How to send program output to a printer or plotter.

## ACHIEVEMENT INDICATORS

1) Define computer literacy in terms of vocational education requirements.
2) List at least five general microcomputer skills which students should possess in a computer-based vocational program.
3) List four steps required in a computer familiarization course to normalize students' microcomputer skills.

## Unit 13

# Evaluate and Modify Computer-Based Instruction Based on Student Achievement

## UNIT OBJECTIVE

Upon completion of this unit, the learner will be able to implement the criterion referenced model of instruction for the purposes of evaluating and modifying CBI.

## SPECIFIC OBJECTIVES

Upon completion of this unit, the learner will be able to:

1) Identify the steps in Glaser's model of criterion referenced instruction.
2) Write objectives for CBI units of instruction.
3) Develop pre- and post-assessment from CBI components of a unit of instruction.
4) Evaluate assessment outcomes and relate the assessment to the effectiveness of CBI in the vocational program.
5) Use task analysis to modify a unit of instruction prior to reteaching the computer-based instructional materials.

# Evaluate and Modify CBI Based on Student Achievement

### BY: DR. MOLLY WILSON-DROTTER

Computer-Based Instruction (CBI) is a relatively novel method of reaching students. It has seemingly met with a great deal of success because computers motivate students and increase their time on task. Yet, CBI has not been thoroughly evaluated as a teaching method in terms of legibility of material displayed on a CRT for poor readers, value of gaming in teaching basic skills, and need for repetition of simple level skills in teaching overall skill concepts.

As accountability becomes more and more prevalent in school districts, it is imperative that teachers are able to prove that CBI—or any chosen method of instruction—is effective. The need to evaluate CBI and to modify it as needed is an essential part of effective educational computing. To date, the instructional measurement model that allows the most flexibility is Criterion Referenced Instruction. Criterion Referenced Instruction lends itself to CBI because its basic concepts can be applied to many different types of computer-based learning activities. It is a flexible measurement and evaluation system.

Criterion Referenced Instruction provides vocational teachers with a method of assessing student learning as it occurs in classrooms and laboratories. Teachers are given the freedom to decide what they want to teach and to assess what they have taught. Thus instruction is geared to learners. Vocational instructors can present material they believe should be covered for specific students while employing instructional methods which they perceive to be appropriate.

Criterion Referenced Instruction, which follows a test-teach-retest format, is designed to be a practical guide to instruction. Two keywords in this type of instruction are **PLANNING** and **TEACHER DECISION MAKING.** This unit, based on Glaser's model of classroom management, shows how to effectively implement both ideas.

## GLASER'S MODEL OF CRITERION—REFERENCED INSTRUCTION

One method of Criterion Referenced Instruction that provides a simple and straightforward approach to classroom evaluation is Glaser's model for teaching (1962).

Glaser's model for classroom instruction provides great flexibility and freedom for the teacher (Tillman, Bersoff, & Dolly, 1976). This model can be used in individualized, group, or whole-class work to meet teacher and student needs in a computer-based vocational program. It includes four steps:

1. Pre-Assessment
2. Behavioral or Instructional Objectives
3. Teaching or Instruction
4. Testing

1. PRE-ASSESSMENT: This step assesses what students know before they begin a particular unit of work or a daily CBI lesson. Pre-assessment can take several forms: (1) A *written pre-test* which determines if students already have the skills to be taught. This step can permit students to skip upcoming work. Additionally, it may determine that a student needs remedial work. (2) An *informal assessment* helps vocational instructors identify certain skills which students need prior to beginning a lesson. It is important for instructors to recognize if students have these skills. If students do not possess prerequisite skills, then instructors should prioritize these skills as the first to be taught. If students have previously mastered these prerequisite skills, this step permits teachers to avoid repetitive learning for students.

   These skills can cover both basic computer literacy skills (presented in Unit C.12) and the vocational skills and knowledge highlighted in this unit of instruction. In CBI, as in other soundly-based forms of instruction, pre-assessment avoids both needless repetition and teaching above the learners' competency levels.

2. BEHAVIORAL or INSTRUCTIONAL OBJECTIVES: Objectives are student goals. They describe what students are expected to learn in particular units or daily lessons. They provide a way for vocational teachers to identify what is to be accomplished prior to the start of the instructional period. Objectives for educational computing should be flexible and capable of being modified. Objectives are starting points for effective computer-based learning.

   In CBI, knowing when to use the computer for assistance with appropriate tasks and how to select students to be trained is facilitated by having goals in mind before instruction begins. Particularly in CBI, clear cut instructional objectives quell the fear of the computer being a play toy for avoiding "real work" in the classroom.

3. TEACHING or INSTRUCTION: This step involves writing lesson plans for CBI. Building on the foundation of objectives, daily planning provides for good teaching. Having lesson plans does not reduce teacher flexibility. Vocational teachers can change lesson plans as the need arises, but having plans provides instructors with starting points.

   Planning is a necessity in CBI, particularly in classrooms where students must share terminals, software, and printers. Lesson planning facilitates classroom management and provides a structure for students.

4. TESTING: The testing step evaluates what students have learned as a result of CBI. This process can be performed daily, at the completion of units of instruction, or as the vocational teacher sees fit. Objectives set forth in Step 2 of the model are converted into test questions. Since this is the material that has been taught in the vocational program, it is the material on which students must be tested. After the test is given, the teacher can evaluate the effectiveness of the instructional unit.

   In cases of failure and in individualized instruction, students falling below the "pass" percentile may redo the unit with modifications to the practice exercises and test questions, and with a variation of instructional methods by which they were taught. CBI is very useful in the recycled instruction phase because it provides individual student practice. Furthermore, it is a handy bookkeeping system for student progress.

   The four steps of Glaser's model of Criterion Referenced Instruction lend themselves to the easy and effective evaluation of CBI. Each step of the Glaser model will be discussed in more detail in the following sections of this unit.

### Pre-Assessments:

There should be two phases of pre-assessment for students in a computer-based vocational program: a) establishing overall goals and b) pre-test.

A. Overall Goals: This phase requires vocational instructors to determine what skills students should be able to perform upon completion of the unit of instruction. Skills, knowledge, and attitudes pertinent to job-specific applications of microcomputers in vocational education should be established. Overall goals are stated in terms of what students will be able to do at the end of the unit of instruction rather than what the teacher will do during the unit. Overall goals are not as clear cut as behavioral objectives, but serve to focus the teacher on the overall crux of the instructional unit. The unit could be one in which CBI is being used as a delivery system or it could be one in which computer skills are being taught.

B. Pre-Test: This is a very important, but simple idea. Vocational teachers must analyze what students will be asked to do in the unit. Then, instructors need to determine which skills students are expected to possess prior to starting the lesson. What are the prerequisites? For example, should they be able to read at a certain level? Should students have certain background knowledge? Will vocational students need certain tools or equipment? What level of computer skills are expected? These background competencies must be present before the student can proceed with the lesson. If they are not, the vocational teacher must provide the backup skills prior to executing the lesson.

Pre-assessment can simplify a great deal because no one begins a unit until all the prerequisite skills are mastered. The teacher has focused goals for each unit and knows beforehand the specific use the computer will have in instruction. Therefore, CBI is a meaningful part of structured instruction, not a gaming device to fill empty instructional hours.

### Instructional Objectives

Instructional objectives are specific student goals—the foundation of good planning. They tell what students will be able to do upon completing a computer-based unit of instruction. Mager (1962) identifies three parts to an objective: a) student behavior, b) testing conditions, and c) performance standards.

A. Student Behavior: Instructional objectives identify what pupils will be able to do when lessons are completed. What students will do must always be something that teachers can observe. For example, assume that an instructor is teaching students how to replace a clutch drive belt. How will the instructor know if students have mastered this skill? Students will be able to list different types of drive belts; identify parts of the pully assembly; perform other types of observable learning activities; and as a concluding activity, demonstrate the ability to replace a clutch drive belt. Often, there are many ways to assess the same skill. The key is that the pupil behavior in the objective has to be an observable action.

B. Testing Conditions: What will students be provided when they are tested on material? If students are tested on tuning an automobile, the conditions of the testing will be very important because they can vary from using an engine analyzer to a feeler gauge. Instructors must decide what types of tasks students should be able to perform. Then, instructors must indicate which tools, equipment, manuals, specifications, or other materials should be used by students to correctly perform the task.

C. Performance Standards: The vocational teacher determines how many answers the student must correctly answer on a post-test to reach mastery. However, advisory committees and industrial standards are often used to establish criterion levels. By establishing standards before the work is done, teachers will know what to expect. This expectation of performance will apply to each student undertaking the unit. Students who fall below the performance standard may repeat a modified unit of instruction which reteaches the subject matter.

The following is an example of an instructional objective in a computer-based vocational program:

Given a microcomputer and software package titled *Micrometer Readings I*, the student will choose micrometer measurements graphically displayed on the CRT by inputting ten correct micrometer readings.

**Student Behavior:** Choose micrometer measurements graphically displayed on the CRT

**Testing Conditions:** Given a microcomputer and software package *Micrometer Readings I*

**Performance Standards:** Inputting ten correct micrometer readings

### Teaching With Lesson Plans

This section of Unit C.13.14 focuses on planning lessons for CBI. Readers are referred to Unit C.3 for a more in-depth look at this topic.

There are many ways to write lesson plans for CBI. This section features a lesson plan format which permits vocational teachers to generate more approaches to teach any one lesson. Teachers can use alternative ideas in case a lesson flops, computers or software malfunction, printers do not print, or some other type of normal catastrophe occurs. Being overly prepared permits vocational teachers to use the best computer activities as opposed to the quickest or most convenient. A sample lesson planning format entails seven components.

I. **Title of the Lesson**

II. **Student Performance Objective(s)**

Teachers should list expected performances for students upon completion of the lesson. These can be computer-based objectives which have been taught for several days or can be objectives for new instructional material.

III. **Entering Behaviors of Students**

What expectations will vocational teachers have for students as they begin a CBI lesson? Instructors need to provide a quick check to determine if students are ready for certain computing activities. If a student does not perform well in a given lesson, this pre-instruction check can help teachers determine if the student needs more practice at the computing activity or if the student did not have the prerequisite skills at the onset of the lesson.

IV. **Methods to Accompany CBI**

Vocational teachers should list several alternative methods for teaching a CBI lesson. Instructors should try to list a range of ideas from the most practical to the most outlandish. This exercise in flexibility serves to provide teachers leeway in ways to teach a lesson. Units C.9, C.10, and C.11 provide expanded views of this topic.

V. **Selecting CBI Methods**

Instructors will find some methods more suitable than others for integrating computer technology into vocational instruction. Instructional methods used with a lesson should be evaluated according to cost; ease of execution; time parameters; availability of hardware; software and other materials; and student reaction to the lesson. Unit C.1 focuses on education applications of microcomputers and provides readers with additional insight for selecting CBI methods.

VI. **Delivering the CBI Lesson**

Vocational instructors should focus on three components of a CBI lesson.

A. The Warm Up: This should be a quick attention-getting technique that will highlight the significance of computer technology in the lesson. This technique prepares students with a "thought set" for the lesson and gets them ready for the upcoming instruction.

B. Teaching the Skill: This phase involves the teacher in using one or several instructional methods for teaching a vocational skill, knowledge, and/or attitude to students. Once again, Unit C.1 provides examples of educational applications of microcomputers in vocational instruction and curricula.

C. Reinforcement: Vocational instructors should create fun or unique activities to help students grasp the content of a lesson. Educational drill and practice or gaming software can be used in this reinforcement phase. Microcomputers can be used extensively in the reinforcement phase of a lesson because of their high motivational value with students.

VII. **Testing**

Vocational instructors should note on lesson plans whether tests will be given at the end of the lesson or at a later date. Software packages may include testing activities or students may have to demonstrate a proficiency with hardware or software. The student instructional objectives provide teachers with the performances and criteria levels for test questions.

## Evaluation

Evaluation is an essential component of an effective computer-based vocational program. Teachers must assess whether students have sufficiently mastered vocational curriculum content in order for them to progress to the other units. Vocational instructors can rely on four considerations for determining a student's mastery of vocational education materials:

A. Asking diverse questions
B. Covering the objectives
C. Using the correct types of test questions
D. Grading

A. Questioning Techinques in Educational Computing: As with traditional modes of instruction, vocational instructors need to request different levels of skill from students engaged in CBI. There is a place for memory work in educational computering, but higher thinking skills must be made a part of CBI to reflect the demands of using computer technology in the work world. Teachers should seek out educational computing materials and software which force students to engage in higher order thinking skills.

The word **taxonomy** means that each learning level builds on the one which precedes it. Vocational instructors can use Bloom's Taxonomy (as an example) for forming questions at a desired level for an educational computing skill, knowledge, or attitude (Bloom, Hastings, & Madaus, 1971).

The levels of Bloom's Taxonomy are:

**Recall** This entails the recognition or memory of specific facts, rules, or principles. Students are asked to recall facts or figures from memory.

**Comprehension** This is the lowest level of understanding. Students are requested to summarize, paraphrase, or make extensions of knowledge.

**Application** Students select a process or rule and correctly apply it to a new situation.

**Analysis** Students take a whole idea and break it down into parts, emphasizing its organization and functions.

**Synthesis** Students create a new pattern or structure. This entails new, creative applications of computer technology to a vocational discipline.

**Evaluation** Students examine their own work or others' and make judgments regarding strengths and weaknesses.

The purpose of a taxonomy in a computer-based vocational program is to encourage students to do more than merely recall answers. Students need opportunities to apply what they learn to create original applications of computer technology in vocational education. These types of creative, analytical skills will pay off for students in the job market.

B. Covering the Objectives: Vocational instructors should make it a practice to test students on all of the performance objectives. If objectives for educational computing are included in lesson plans and units of instruction, then tests should be prepared to assess students' mastery of those objectives.

Objectives and test items should test students at the same level of the taxonomy as a skill was taught. In other words, an objective should not be written at the recall level and a corresponding test item constructed at the analysis level. Vocational instructors can use a table of specifications to graphically display the levels of the taxonomy at which the objectives and test items are written, and to see how well they mesh with the instructional goals of the computer-based instructional program.

## TABLE OF SPECIFICATIONS

### SPECIFIC OBJECTIVES OF COMPUTER-BASED VOCATIONAL PROGRAM

| CBI Instructional Evaluation Goals | Recall | Comprehension | Application | Analysis |
|---|---|---|---|---|
| Instructional Goal A | | Objective No. 1 | Objective No. 2 | |
| Instructional Goal B | Objective No. 4 | | | Objective No. 3 |
| Instructional Goal C | | Objective No. 5 | Objective No. 6 | |

C. Using the Correct Types of Test Questions: Vocational instructors realize that different types of test questions test different levels of knowledge (Gay, 1980; Gronlund, 1976; Payne, 1974). Certain test items mesh well with select educational applications of microcomputers. Other types of test items may lend themselves to the strengths of computer managed instruction (See Unit E.6.7.8).

True-false questions permit the use of a greater number of questions, but they are used predominantly at the recall and comprehension levels. Multiple choice and matching questions eliminate much guessing of answers by students. These types of questions are difficult to write because they require teachers to formulate four or five alternative answers. However, well written multiple choice and matching questions test above the recall level of Bloom's Taxonomy.

True-false, multiple choice, and matching test items permit computer scoring and evaluating, as discussed in Unit E.6.7.8. Short answer test items require students to recall information and formulate a response. Vocational instructors have the responsibility of judging the correctness of the response. Essay items provide students opportunities to organize information and present it in their own words. Students learn to collect their thoughts and express themselves. However, essay tests are time consuming to grade and they penalize students who are not good writers. Additionally, essay tests restrict the number of questions which can be asked in an allotted time period.

D. Grading: Vocational instructors will encounter dilemmas with grading in CBI similar to those they have encountered with traditional instructional approaches. Teachers who apply the techniques of mastery learning eventually face the decision of what letter grade to assign to mastery. Vocational schools and programs which feature competency-based education have taken various approaches to this issue. Students can be graded pass/fail, letter grades can be assigned to varying competency levels for each task, or students can all receive A's once they meet mastery levels. Some programs and schools do not use letter grades. Instead, progress charts present numerical values for competency levels for all of the tasks which make up the instructional program.

Teachers must decide upon a grading practice for the vocational program, set standards, and inform students of what is expected of them to reach mastery level. It is imperative that students be informed of these standards prior to the beginning of instruction.

Those who do not meet the criterion level are retaught the materials, with modifications in practice exercises and test questions. Computer-Based Instruction is especially useful in remediating the skills of students independently while other students engage in other units of instruction.

### Recycling Students

For students who do not reach competency, the vocational teacher must make decisions about reteaching material to criterion. These decisions are based on selecting objectives that were not mastered, analyzing why they were not mastered, and modifying instruction to reteach necessary concepts.

If a teacher has worked through the criterion referenced model of test construction, each test question should clearly fit an objective for the unit. Teachers can easily examine student papers, projects, or software responses to assess which objectives were not met. Objectives on which the student mastered fewer concepts than the teacher designated in the performance standard should be retaught.

For most test items, teachers can examine the incorrect answers chosen and reconstruct the misinformation the student was using in giving answers. The misinformation must be the focus of the reteaching of material. At times, the best way to comprehend wrong answers is to simply ask the students why they chose a particular answer. Clear-cut fallacies can be remediated rather simply. Guessing answers randomly will require reteaching the entire concept.

If the objective involves a complex, sequential task, a task analysis is in order for deciding what skills to remediate. Task analysis involves the teacher working through a particular task and noting the subskills involved. Then, in order to assess what the student needs to have remediated, the teacher observes the student work through the task, checking off each mastered subskill in the sequence. Only unmastered steps are remediated until they are mastered.

After the teacher decides on the objectives to be remediated and the particular skills to be retaught, instructional strategies must be modified to reteach material. Computer-Based Instruction is an extremely useful tool for reteaching because it allows the teacher to give individualized remediation to those who need it while concentrating his or her efforts on instructing the other students who are ready to move ahead.

In the reteaching phase, the teacher must construct new practice exercises and, later, new test questions—all matching the objective ano concentrating on the misconceptions that were identified from examining the test. A data bank of practice exercises and test items will facilitate the remediation process for a teacher using the Criterion Referenced Instruction model. Again, the use of a computer memory bank is an efficient way to store data. Programs can be written to randomize practice items and to call up items that relate to specific objectives. Students can receive feedback on correct and incorrect responses, and that data can be stored for the teacher, so that he or she will know when to retest the student on concepts being remediated.

When the student is ready for the retest, the teacher may choose to readminister the entire test again, or to only retest the objectives that were not passed initially. Test items on the retest should be keyed to specific objectives, but should present new information to the student so that he or she is not merely retaking the original test. A computer terminal may be used to administer the test so that the student is working individually and quietly. The computer can score the student's retest and let the student and teacher know if the objectives and material have been mastered.

## SUMMARY

Practicing vocational instructors have already realized that teaching involves much work. Planning is the key to success for CBI just as it has been for traditional modes of instruction. Computer-Based Instruction is based on a teach-test-reteach model. An advantage of CBI compared to other forms of instruction is that it provides students with a motivating method to practice new skills. Computing activities can be good ways to remediate students who have not mastered vocational materials. These features, in addition to computer managed instruction features (Unit E.6.7.8), make CBI a welcome component of vocational curricula and instruction.

## ACHIEVEMENT INDICATORS

1) Identify the steps in Glaser's model of criterion referenced instruction.

2) Write objectives for CBI units of instruction.

3) Develop pre- and post-assessments for CBI components of a unit of instruction.

4) Evaluate assessment outcomes and relate the assessment to the effectiveness of CBI in the vocational program.

5) Use task analysis to modify a unit of instruction prior to reteaching the computer-based instructional materials.

## REFERENCES

Bloom, B. S., Hastings, J. T., & Madaus, G. F. (1971). *Handbook on Formative and Summative Evaluation.* New York: McGraw-Hill.

Gay, L. R. (1980). *Educational Evaluation and Measurement.* Columbus, OH: Merrill.

Glaser, R. M. (1962). Psychology and Instructional Technology. In R. Glaser (Ed.), *Training Research and Education.* Pittsburgh: University of Pittsburgh Press.

Gronlund, N. E. (1976). *Measurement and Evaluation in Teaching.* New York: MacMillan.

Mager, R. F. (1962). *Preparing Instructional Objectives.* New York: Fearon.

Payne, D. A. (1974). *The Assessment of Learning: Cognitive and Affective.* Boston: D. C. Heath.

Tillman, M., Bersoff, D., & Dolly, J. (1976). *Learning to Teach: A Decision Making System.* Boston: D. C. Heath.